

FUZZY CONTROLLER DESIGN FOR PARAMETRIC CONTROLLERS

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND
ELECTRONICS ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Murat Akgöl
September 1996

71.615

QA

9.64

.A34

1996

FUZZY CONTROLLER DESIGN FOR PARAMETRIC CONTROLLERS

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND
ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By

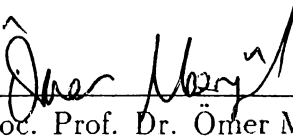
Murat Akgül

September 1996

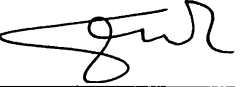
QA
9.64
.A34
1896

B.035240

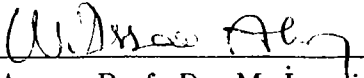
I certify that I have read this thesis and that in my opinion it is fully adequate,
in scope and in quality, as a thesis for the degree of Master of Science.


Assoc. Prof. Dr. Ömer Morgül(Supervisor)

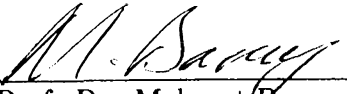
I certify that I have read this thesis and that in my opinion it is fully adequate,
in scope and in quality, as a thesis for the degree of Master of Science.


Prof. Dr. M. Erol Sezer

I certify that I have read this thesis and that in my opinion it is fully adequate,
in scope and in quality, as a thesis for the degree of Master of Science.


Assoc. Prof. Dr. M. İrşadi Aksun

Approved for the Institute of Engineering and Sciences:


Prof. Dr. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

FUZZY CONTROLLER DESIGN FOR PARAMETRIC CONTROLLERS

Murat Akgül

M.S. in Electrical and Electronics Engineering

Supervisor: Assoc. Prof. Dr. Ömer Morgül

September 1996

In this thesis, fuzzy logic controller (FLC) design for tuning some parametric controller is investigated. The objective in designing an FLC is to determine the rule base of the system and the data base which includes membership functions, set operations, and inference engine. Two designs have been realized using heuristic rule generation; one for a PID controller and one for a lead-lag type controller. The FLCs in these designs set the parameters of the PID and lead-lag controller on-line. The rules and the corresponding membership functions are constructed by observing the effect of the changes of the parameters on the overall performance. One other design is performed using the desired input-output data pairs. In this design Fuzzy c-Means clustering algorithm is used to extract the rules and the membership functions from the input-output data of the system. Simulation results showed that better controller performance can be achieved by FLCs in comparison with the classical design methods.

Keywords : Fuzzy logic, fuzzy logic controllers, fuzzy c-means clustering algorithm, membership function.

ÖZET

PARAMETRİK DENETLEYİCİLER İÇİN BULANIK DENETLEYİCİ TASARLANMASI

Murat Akgül

Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans

Tez Yöneticisi: Assoc. Prof. Dr. Ömer Morgül

Eylül 1996

Bu tezde, parametrik denetleyicilerin ayarlanması için bulanık mantığa dayalı denetleyici (BMDD) tasarımı incelenmiştir. Bulanık mantığa dayalı denetleyici tasarımında amaç, sistemin kural tabanını ve içinde üyelik fonksiyonlarının, küme işlemlerinin, ve çıkarım mekanizmasının bulunduğu veri tabanının belirlenmesidir. Biri PID diğeri faz kaydırmalı denetleyici için sezgiye dayalı iki tasarım gerçekleştirilmiştir. Bu tasarımdaki BMDD'ler PID ve faz kaydırmalı denetleyicilerin parametrelerini eş zamanlı olarak belirlemektedir. Kurallar ve onlara bağlı üyelik fonksiyonları, bu parametre değişimlerinin bütün sistemin performansına etkisi gözlenerek oluşturulmuştur. Bir başka tasarım da sisteme ait istenen giriş ve çıkış sinyalleri kullanılarak gerçekleştirilmiştir. Bu tasarımda, Bulanık c-Ortalamalı kümeleme algoritması, kuralların ve üyelik fonksiyonların eldeki giriş ve çıkış sinyallerinden çıkartılmasında kullanılmıştır. Simulasyon sonuçları, BMDD'ler ile klasik tasarım metodlarına göre daha iyi bir performansa ulaşılabilirliğini göstermiştir.

Anahtar kelimeler : Bulanık mantık, bulanık mantık denetleyiciler, bulanık c-ortalamalı kümeleme algoritması, üyelik fonksiyonu.

ACKNOWLEDGMENTS

I would like to express my deep gratitude to my supervisor Assoc. Prof. Dr. Ömer Morgül for their guidance, suggestions, and invaluable encouragement throughout the development of this thesis. I would like to thank to Prof. Dr. M. Erol Sezer and Assoc. Prof. Dr. M. İrşadi Aksun for reading and commenting on the thesis.

And special thanks to all Graduate Students in this department for their continuous support.

TABLE OF CONTENTS

| | | |
|----------|---|-----------|
| 1 | INTRODUCTION | 1 |
| 2 | Mathematical Preliminaries | 4 |
| 2.1 | Fuzzy Sets | 4 |
| 2.2 | Operations On Fuzzy Sets | 5 |
| 2.3 | Fuzzy Relations | 6 |
| 2.4 | Implication | 10 |
| 3 | FUZZY CONTROL | 12 |
| 3.1 | The Control Problem | 12 |
| 3.2 | Classification Of Fuzzy Logic Systems | 13 |
| 3.2.1 | Pure FLS | 13 |
| 3.2.2 | Takagi and Sugeno's FLS | 14 |
| 3.2.3 | FLS with Fuzzifier and Defuzzifier | 14 |
| 3.3 | Computational Structure of FLS with Fuzzifier and Defuzzifier . | 15 |
| 3.3.1 | Fuzzification | 17 |
| 3.3.2 | Rule Firing | 19 |

| | | |
|----------|---|-----------|
| 3.3.3 | Defuzzification | 20 |
| 3.4 | Design of Fuzzy Logic Controller | 20 |
| 3.4.1 | Derivation of Rules | 22 |
| 4 | Fuzzy Logic Controller Design : Heuristic Rule Generation | 24 |
| 4.1 | PID Controller Design | 24 |
| 4.1.1 | Rule-Base Construction | 25 |
| 4.1.2 | FLC Structure | 28 |
| 4.1.3 | Simulations and Results | 29 |
| 4.2 | Lead-Lag Controller Design | 34 |
| 4.2.1 | Rule-Base Construction | 34 |
| 4.2.2 | FLC Structure | 37 |
| 4.2.3 | Simulations and Results | 37 |
| 5 | Fuzzy Controller Design : Rule generation using fuzzy c-means clustering algorithm | 40 |
| 5.1 | Dynamic Programming | 43 |
| 5.2 | Fuzzy C-Means Clustering Algorithm | 44 |
| 5.3 | Formation of Membership Functions | 46 |
| 5.4 | Simulations and Results | 47 |
| 6 | Conclusion | 51 |

LIST OF FIGURES

| | | |
|-----|---|----|
| 3.1 | System structure under investigation | 12 |
| 3.2 | Basic configuration of pure FLS | 14 |
| 3.3 | Basic configuration of Takagi and Sugeno's FLS | 14 |
| 3.4 | FLS with fuzzifier and defuzzifier | 15 |
| 3.5 | Computational Structure of FLS with fuzzifier and defuzzifier . | 15 |
| 3.6 | Membership functions | 18 |
| 3.7 | Clipped fuzzy sets as a result of rule firing | 19 |
| 4.1 | System structure under investigation | 25 |
| 4.2 | A typical step response of a system. | 26 |
| 4.3 | Membership functions for error and increment of error | 28 |
| 4.4 | Membership function for the variables K_p , K_i , and K_d | 29 |
| 4.5 | Membership function for the variables K_p , K_i , and K_d | 31 |
| 4.6 | Comparison of the step responses. | 32 |
| 4.7 | PID parameters determined by the FLCs. | 32 |

| | | |
|------|--|----|
| 4.8 | Comparison of the step responses. | 33 |
| 4.9 | PID parameters determined by the FLCs. | 33 |
| 4.10 | System structure under investigation | 34 |
| 4.11 | Step responses of $P_1(s)$ for $a=0.01, 0.3, 0.7, 1, 2, 5, 10$. | 35 |
| 4.12 | Step responses of $P_2(s)$ for $a=0.01, 0.3, 0.7, 1, 2, 3, 4$. | 35 |
| 4.13 | Membership functions of the linguistic variable a | 36 |
| 4.14 | Comparison of the step responses of $P_3(s)$ with the two different controller. | 38 |
| 4.15 | (a) Step responses of $P_4(s)$. (b) Variation of the parameter a . | 39 |
| 5.1 | The membership functions and the relation of the rule if “e is LE_i ” then “u is LU_i ”. | 41 |
| 5.2 | Negative unit feedback with a parametric controller $C(s, \mathbf{p})$. . . | 41 |
| 5.3 | Sample point of error versus the corresponding control signal. | 42 |
| 5.4 | Negative unit feedback with a proportional controller K | 47 |
| 5.5 | The suboptimum solution for the proportional control and the corresponding step response. | 47 |
| 5.6 | Rule base of the fuzzy controller | 49 |
| 5.7 | (a) The step responses with two different controller (b) The waveform for the proportional gain. | 50 |
| 5.8 | (a) The step responses with two different controller for the system with disturbed parameter (b) The waveform for the proportional gain. | 50 |

To My Family

Chapter 1

INTRODUCTION

Over the past few years many researchers of various fields and manufacturers of control equipment pay a great attention to fuzzy control. There are still however, two predominant extreme positions as to the benefits of fuzzy control. On the one hand, many proponents of this technology claim that fuzzy control will revolutionize control engineering, promises major breakthroughs, and will be able to solve complex engineering problems with very little effort. On the other hand, many representatives of the control engineering community still proclaim that “everything that can be done in fuzzy control can be done conventionally as well” and announce a breakdown of the “fuzzy hype” in the near future.

A fuzzy control system is a real time expert system implementing a part of a human operator’s or process engineer’s expertise which does not lend itself to being easily expressed in PID-parameters of differential equation but rather in situation/action rules. However, fuzzy control differs from main-stream expert system technology in several aspects. One main feature of fuzzy control systems is that there are symbolic if-then rules and qualitative, fuzzy variables and values such as

if “pressure is high” and “slightly increasing” then “energy supply is negative medium”

Most of the inventors of fuzzy control have a strong control engineering and systems theory background. From their perspective, fuzzy control can be seen as heuristic and modular way for defining nonlinear, table based systems. Reconsider the rule above : it is nothing but an informal “nonlinear PD-element”.

A representation theorem, mainly due to Kosko [1], states that any continuous nonlinear function can be approximated as closely as needed with a finite set of fuzzy variables, values, and rules. This theorem describes the representational power of fuzzy control in principle, but it does not answer the questions of how many rules are needed and how they can be found, which are of course essential to real-world problems and solutions.

It is generally agreed that an important point in the evolution of the modern concept of uncertainty was the publication of a seminal paper by Lofti A. Zadeh [1965], even though some ideas presented in the paper were envisioned some 30 years earlier by the American philosopher Max Black [1937]. In his paper, Zadeh introduced a theory whose objects-*fuzzy sets*-are sets with boundaries that are not precise. The membership in a fuzzy set is not a matter of affirmation or denial, but rather a matter of *degree*.

The literature in fuzzy control has been growing rapidly in recent years, making it difficult to present a comprehensive survey of the wide variety of applications that have been made. Historically, the important milestones in the development of fuzzy control may be summarized as shown in Table.1.1[2].

The first successful industrial application of the FLC was a cement kiln control system developed by the Danish cement plant manufacturer F. L. Smidth in 1979.

Fuzzy Logic Controllers(FLCs) can be used in various ways. One of them is to use the FLC directly as the controller for the system under consideration. The other choice is to use a standard controller like PID or lead-lag and tune the parameters of this controller with a fuzzy logic system (FLS). A few of the studies on parameter tunings are [3], [4] and [5]. In [4] the PID parameters are tuned according to some rules derived by considering a typical system output. In [5] some changes to the model of fuzzy PI controller is introduced to reduce

| | | |
|------|--------------------------|--|
| 1972 | Zadeh | A rationale for fuzzy control |
| 1973 | Zadeh | Linguistic approach |
| 1974 | Mamdani & Assilian | Steam engine control |
| 1976 | Rutherford <i>et al.</i> | Analysis of control algorithms |
| 1977 | Ostergaard | Heat exchanger and cement kiln control |
| 1977 | Willaeys <i>et al.</i> | Optimal fuzzy control |
| 1979 | Komolov <i>et al.</i> | Finite automation |
| 1980 | Tong <i>et al.</i> | Wastewater treatment process |
| 1983 | Takagi and Sugeno | Derivation of fuzzy control rules |
| 1984 | Sugeno and Murakami | Parking control of a model car |
| 1985 | Togai and Watanabe | Fuzzy chip |
| 1986 | Yamakawa | Fuzzy controller hardware system |
| 1988 | Dubois and Prade | Approximate reasoning |

Table 1.1: Some important studies in fuzzy control.

the overshoot even more.

In this thesis the implementation of a fuzzy controller which sets the parameters of a parametric controller has been investigated. Mainly two different methods have been used in constructing the rule base. In the first method we obtain the rules by observing the input output data of the system. Here we obtain the fuzzy rules to set the parameters of the parametric controller heuristically. For the controller, we consider the standard PID, and the lead-lag type controllers. In the other part again the input-output data is used but this time the rules have been set from the results of Fuzzy C-Means (FCM) clustering algorithm.

In Chapter 2 some definitions and the mathematical preliminaries to fuzzy operations on fuzzy sets are given. In Chapter 3 an introduction to fuzzy control is given and various ways to find the required rules are explained. In the remaining chapters fuzzy controller design for PID and lead-lag type controllers are investigated. In Chapter 4 the rule generation is done heuristically and in Chapter 5 this is performed by Fuzzy C-Means(FCM) clustering algorithm. Finally in Chapter 6 we give some concluding remarks.

Chapter 2

Mathematical Preliminaries

2.1 Fuzzy Sets

A classical set is a collection of objects of any kind. Let X be the universe of discourse on which the set \mathcal{A} is defined. There are several ways to decide if an element x belongs to the set \mathcal{A} or not. One way is to use a characteristic function.

Definition 1 $\mu_{\mathcal{A}} : X \longrightarrow \{0, 1\}$ is a characteristic function of the set \mathcal{A} iff

$$\forall x \in X, \quad \mu_{\mathcal{A}} = \begin{cases} 1 & , x \in \mathcal{A} \\ 0 & , x \notin \mathcal{A}. \end{cases} \quad \square$$

The characteristic function $\mu_{\mathcal{A}}$ takes values 1 and 0 depending on whether an element belongs to the set \mathcal{A} or not. This is due to the lack of ambiguity in the definition of set \mathcal{A} . Let \mathcal{B} be a set defined as “numbers close to 1.5”. Here the closeness in the definition is not clear. Due to this vagueness the set \mathcal{B} can not have sharp borders as the set \mathcal{A} has. Therefore one can not use the codomain $\{0, 1\}$ of the characteristic function, instead for any element x a grade of membership value in the interval $[0, 1]$ must be assigned. After this motivation the definition of fuzzy set can be done as follows;

Definition 2 *The membership function $\mu_{\mathcal{F}}$ of a fuzzy set \mathcal{F} is a function $\mu_{\mathcal{F}} : U \longrightarrow [0, 1]$. \mathcal{F} is completely determined by the set of tuples*

$$\mathcal{F} = \{(\mu_{\mathcal{F}}(u), u) | u \in U\} \square$$

Let us reconsider the fuzzy set \mathcal{B} , “numbers close to 1.5”. This fuzzy set can be represented with the following set of tuples ;

$$\mathcal{B} = \{(0.0, 0.0), (0.2, 0.5), (0.5, 1.0), (1.0, 1.5), (0.5, 2.0), (0.2, 2.5), (0.0, 3.0)\}.$$

Here the first entry of the tuples is the grade of membership of the second entry, u . These tuples are also given with the following representation:

$$\mathcal{B} = \{0.0/0.0, 0.2/0.5, 0.5/1.0, 1.0/1.5, 0.5/2.0, 0.2/2.5, 0.0/3.0\}.$$

The division symbol, in the above notation, does not mean to divide the membership grade to the value of the element u , it is just used to represent the pairs.

2.2 Operations On Fuzzy Sets

In classical set theory, the union, intersection and complement of sets are simple operations that are unambiguously defined. The unambiguity follows from the fact that logical operators *AND*, *OR*, and *NOT* have a well defined semantics based on propositional logic. For example, “ \mathcal{A} *AND* \mathcal{B} ” in propositional logic is true if and only if both expressions \mathcal{A} and \mathcal{B} are true. In fuzzy set theory their interpretation is not so simple, because graded characteristic functions (membership function) are used. Zadeh proposed [6] the following definitions for the union, intersection and complement operations on fuzzy sets.

Definition 3 *Let \mathcal{A} and \mathcal{B} be two fuzzy sets and let $\mu_{\mathcal{A}}(\cdot)$ and $\mu_{\mathcal{B}}(\cdot)$ be their membership functions, respectively. Then the membership function of the sets*

$\mathcal{A} \cup \mathcal{B}$, $\mathcal{A} \cap \mathcal{B}$, \mathcal{A}^c may be given as follows :

$$\begin{aligned}\forall x \in X : \mu_{\mathcal{A} \cap \mathcal{B}}(x) &= \min(\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x)), \\ \forall x \in X : \mu_{\mathcal{A} \cup \mathcal{B}}(x) &= \max(\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x)), \\ \forall x \in X : \mu_{\mathcal{A}^c}(x) &= 1 - \mu_{\mathcal{A}}(x). \quad \square\end{aligned}$$

If the values of $\mu_{\mathcal{A}}(x)$ and $\mu_{\mathcal{B}}(x)$ are restricted to the set $\{0, 1\}$ then the results reduce to the classical set operations. Therefore this is a very simple extension of the classical operations. There are other extensions, for example,

$$\begin{aligned}\forall x \in X : \mu_{\mathcal{A} \cap \mathcal{B}}(x) &= \mu_{\mathcal{A}}(x) \cdot \mu_{\mathcal{B}}(x), \\ \forall x \in X : \mu_{\mathcal{A} \cup \mathcal{B}}(x) &= \min(1, \mu_{\mathcal{A}}(x) + \mu_{\mathcal{B}}(x)).\end{aligned}$$

In comparison with the lattice (min and max) operations, here the degree of membership depends on both the values of membership function $\mu_{\mathcal{A}}(x)$ and $\mu_{\mathcal{B}}(x)$. In our studies the operations proposed by Zadeh was used. A list of other operations can be found in [2], [7] and [8].

Among these operations, t-norm and t-conorm(sometimes referred as s-norm) take many researchers interest. These logical connectives posses some properties like boundary conditions, commutativity, associativity. The boundary conditions indicate that the logical connectives for fuzzy sets coincide with those applied in two valued logic. The property of commutativity reflects a case in which the truth value of a composite expression does not depend on the order of the components used in its formation. These concepts are explained in [7] and [8] in more detail with examples.

2.3 Fuzzy Relations

A relation can be considered as a set of tuples, where a tuple is an ordered pair. A binary tuple is denoted as (x, y) , and in general an n-ary tuple is (x_1, x_2, \dots, x_n) . Just like classical sets, classical relations can be described by a characteristic function as well.

Definition 4 $\mu_{\mathcal{R}} : X_1 \times X_2 \times \cdots \times X_n \longrightarrow \{0, 1\}$ is a characteristic function of the set \mathcal{R} iff

$$\forall (x_1, x_2, \dots, x_n) \in X_1 \times X_2 \times \cdots \times X_n, \quad \mu_{\mathcal{R}} = \begin{cases} 1 & , (x_1, x_2, \dots, x_n) \in \mathcal{R} \\ 0 & , (x_1, x_2, \dots, x_n) \notin \mathcal{R}. \quad \square \end{cases}$$

In a fuzzy relation this characteristic function is extended to the interval $[0, 1]$. A fuzzy relation is a fuzzy set of tuples, i.e. each tuple has a membership degree between 0 and 1.

Definition 5 Let U and V be uncountable (continuous) universes and $\mu_{\mathcal{R}} : U \times V \longrightarrow [0, 1]$. Then

$$\mathcal{R} = \int_{U \times V} \mu_{\mathcal{R}}(u, v)/(u, v),$$

is a binary relation on $U \times V$. If U and V are countable (discrete) universes, then we can take

$$\mathcal{R} = \sum_{U \times V} \mu_{\mathcal{R}}(u, v)/(u, v). \quad \square$$

In this notation the division is used to represent the pairs of an element and its membership grade of the corresponding fuzzy set. The integral and the summation symbols do not mean the well known algebraic integration over an argument or taking the sum of all the pairs. They are used to combine all the pairs which constitute the set \mathcal{R} . In this thesis only the discrete universes are studied. Therefore from now on, only the second part of the definition will be used.

Fuzzy relations are very important in fuzzy control because they can describe interactions between variables. Union, intersection and complement operations can be defined in any way as was described in the previous section. Apart from these operations there are two very important operations on fuzzy sets and fuzzy relation, namely projection and cylindrical extension.

Definition 6 Let U and V be two independent domain. And let \mathcal{R} be a fuzzy relation on $U \times V$. Then the projection of \mathcal{R} on V is defined by

$$\text{proj } \mathcal{R} \text{ on } V = \sum_V \max_{x_{j_1}, \dots, x_{j_l}} \mu_{\mathcal{R}}(x_1, \dots, x_n) / (x_{i_1}, \dots, x_{i_k}). \square$$

Example : Let X and Y be two independent domains such that a relation \mathcal{R} is defined on $X \times Y$. Assume that the relation \mathcal{R} is defined as “ x is considerable larger than y ”. Then this relation may be given by the following array :

| \mathcal{R} | y_1 | y_2 | y_3 | y_4 |
|---------------|-------|-------|-------|-------|
| x_1 | 0.8 | 1.0 | 0.1 | 0.7 |
| x_2 | 0.0 | 0.8 | 0.0 | 0.0 |
| x_3 | 0.9 | 1.0 | 0.7 | 0.8 |

Then the projection of \mathcal{R} on X means that x_i is assigned the highest membership degree from the tuples (x_i, y_j) , $j=1, \dots, 4$ for $i=1, \dots, 3$. So,

$$\text{proj } \mathcal{R} \text{ on } X = 1.0/x_1 + 0.8/x_2 + 1.0/x_3,$$

$$\text{proj } \mathcal{R} \text{ on } Y = 0.9/y_1 + 1.0/y_2 + 0.7/y_3 + 0.8/y_4.$$

Definition 7 Let \mathcal{S} be a fuzzy set defined on V , and let U be a domain independent of V . Then the cylindrical extension of \mathcal{S} into U is a relation on $U \times V$ defined as:

$$ce(\mathcal{S}) = \sum_U \mu_{\mathcal{S}}(x_{i_1}, \dots, x_{i_k}) / (x_1, \dots, x_n).$$

, where k and n are the dimensions of the domains V and $U \times V$, respectively.

□

Example : Consider the fuzzy sets \mathcal{A} and \mathcal{B} given below

$$\mathcal{A} = \text{proj } \mathcal{R} \text{ on } X = 1.0/x_1 + 0.8/x_2 + 1.0/x_3,$$

$$\mathcal{B} = \text{proj } \mathcal{R} \text{ on } Y = 0.9/y_1 + 1.0/y_2 + 0.7/y_3 + 0.8/y_4.$$

So the cylindrical extension of \mathcal{A} into Y and the cylindrical extension of \mathcal{B} into X are given as follows :

| $ce(\mathcal{A})$ | y_1 | y_2 | y_3 | y_4 | $ce(\mathcal{B})$ | y_1 | y_2 | y_3 | y_4 |
|-------------------|-------|-------|-------|-------|-------------------|-------|-------|-------|-------|
| x_1 | 1.0 | 1.0 | 1.0 | 1.0 | x_1 | 0.9 | 1.0 | 0.7 | 0.8 |
| x_2 | 0.8 | 0.8 | 0.8 | 0.8 | x_2 | 0.9 | 1.0 | 0.7 | 0.8 |
| x_3 | 1.0 | 1.0 | 1.0 | 1.0 | x_3 | 0.9 | 1.0 | 0.7 | 0.8 |

Having defined cylindrical extension, one can calculate the intersection of a set \mathcal{A} on X and a relation \mathcal{R} on $X \times Y$ by first taking the cylindrical extension of \mathcal{A} into Y and taking the minimum of the associated elements of the two fuzzy sets, $ce(\mathcal{A})$ and \mathcal{R} , where both are defined on $X \times Y$.

The combination of a fuzzy set and a fuzzy relation with the aid of cylindrical extension and projection is called a *composition*. It is denoted by the symbol “ \circ ”. A fuzzy relation, as its name indicates, is a relation between its subsets. This operation is necessary when one knows the relation of a certain subject and some related facts and wants to calculate the facts which he does not know.

Definition 8 Let \mathcal{A} be a fuzzy set defined on X and \mathcal{R} be a fuzzy relation defined on $X \times Y$. Then the composition of \mathcal{A} and \mathcal{R} resulting in a fuzzy set \mathcal{B} defined on Y is given by

$$\mathcal{B} = \mathcal{A} \circ \mathcal{R} = \text{proj} (ce(\mathcal{A}) \cap \mathcal{R}) \text{ on } Y. \square$$

Example : Consider a relation defined as “ x is approximately equal to y ”, \mathcal{R} , defined on $X \times Y$ and suppose it is known that “ x is small” is expressed by a fuzzy set \mathcal{A} on X . Let the fuzzy relation \mathcal{R} , and the fuzzy set \mathcal{A} be given as follows,

| \mathcal{R} | y_1 | y_2 | y_3 |
|---------------|-------|-------|-------|
| x_1 | 1.0 | 0.8 | 0.3 |
| x_2 | 0.8 | 1.0 | 0.8 |
| x_3 | 0.3 | 0.8 | 1.0 |

$$\mathcal{A} = 0.2/x_1 + 1.0/x_2 + 0.8/x_3.$$

Then the cylindrical extension of \mathcal{A} into Y is,

| $ce(\mathcal{A})$ | y_1 | y_2 | y_3 |
|-------------------|-------|-------|-------|
| x_1 | 0.2 | 0.2 | 0.2 |
| x_2 | 1.0 | 1.0 | 1.0 |
| x_3 | 0.8 | 0.8 | 0.8 |

So the intersection of $ce(\mathcal{A})$ and \mathcal{R} , calculated with the minimum operator, results in the following relation :

| $ce(\mathcal{A}) \cap \mathcal{R}$ | y_1 | y_2 | y_3 |
|------------------------------------|-------|-------|-------|
| x_1 | 0.2 | 0.2 | 0.2 |
| x_2 | 0.8 | 1.0 | 0.8 |
| x_3 | 0.3 | 0.8 | 0.8 |

And finally taking the projection into Y results

$$\mathcal{B} = 0.8/y_1 + 1.0/y_2 + 0.8/y_3.$$

The fuzzy set \mathcal{B} can be characterized as “ x is approximately equal to y ” *AND* “ x is small” or equivalently “ y is more or less small”.

2.4 Implication

In classical logic the truth value of $p \Rightarrow q$ is defined as shown in the Table.2.1. This implication or if-then (if p then q) statement can be expressed with different combination of union, intersection and complement operations.

Some equivalent expressions would be $\neg p \vee q$ and $(p \wedge q) \vee \neg p$ where “ \vee ”, “ \wedge ” and “ \neg ” stand for union, intersection and complement operations, respectively. Here p and q can be considered as sets which are defined by a characteristic function whose domain is the set $\{0, 1\}$. To extend the implication operation into fuzzy domain again the codomain of the characteristic function of p and q must be extended to the closed unit interval, $[0,1]$, and instead of the classical

| p | q | $p \Rightarrow q$ |
|---|---|-------------------|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Table 2.1: Truth table for p implies q

set operations, their fuzzy counterparts must be used. Having done these, one can calculate the truth value of an if-then statement. A typical example which may be encountered in fuzzy control might be the following type of statement,

if “pressure is rather high” then “energy supply is negative big”.

The classical set operators $p \Rightarrow q$, $\neg p \vee q$, and $(p \wedge q) \vee \neg p$, have all the same truth table. When these operators are replaced with fuzzy operators, then the equivalent of $p \Rightarrow q$ may differ according to the selected union, intersection and complement operators. There are a number of relations that can be used to represent the meaning of an if-then statement. Some of these are equivalent to the logical implication $p \Rightarrow q$ and some are not. In this study Mamdani implication defined below is used.

Suppose that there is an if-then statement as given below and it is desired to calculate the truth value of this implication.

if \mathcal{A} then \mathcal{B} .

Let \mathcal{A} and \mathcal{B} be two fuzzy proposition defined on X and Y, respectively. Assume that \mathcal{R}_c be the relation defined with the above if-then statement, then the calculation related with the Mamdani implication is as follows,

$$\mathcal{R}_c = ce(\mathcal{A}) \wedge ce(\mathcal{B}),$$

$$\mu_{\mathcal{R}_c}(x, y) = \min(\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(y)).$$

There are other implication operations. An extended list of these can be found in [2], [7].

Chapter 3

FUZZY CONTROL

3.1 The Control Problem

The control problem which we tried to investigate in this thesis is reference input tracking. The system structure under investigation is unit feedback with controller and linear systems which is shown in Fig.3.1

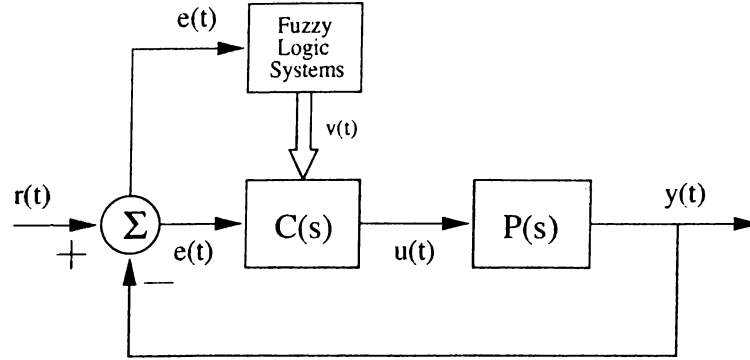


Figure 3.1: System structure under investigation

In this figure, $C(s)$ is the transfer function of a parametric controller, $P(s)$ is the transfer function of the plant under investigation, $r(t)$ is the reference input, $y(t)$ is the output of the system, $u(t)$ is the output of the controller,

$e(t)=r(t)-y(t)$ is the input of the controller and, $v(t)$ is a vector function in \Re^w representing the waveform for the parameters of the controller. Here w is the number of parameters that the controller $C(s)$ has. These parameters are updated by the Fuzzy Logic System (FLS) according to the error signal. In this thesis the problem is investigated for the unit step reference signal.

The block Fuzzy Logic Systems in Fig.3.1 may be composed of more than one FLS, described in the following section. For each parameter of the controller there is a corresponding FLS with its own rules and membership functions.

3.2 Classification Of Fuzzy Logic Systems

Fuzzy logic systems is a name for the systems which have a direct relationship with fuzzy concepts (like fuzzy sets, linguistic variables, and so on) and fuzzy logic. The most popular fuzzy logic systems in the literature may be classified into three types: pure fuzzy logic systems, Takagi and Sugeno's fuzzy system, and fuzzy logic systems with fuzzifier and defuzzifier.

3.2.1 Pure FLS

The basic configuration for a pure FLS is shown in Fig.3.2, where the fuzzy rule base consists of a collection of fuzzy if-then rules, and the fuzzy inference engine uses these fuzzy if-then rules to determine a mapping from fuzzy sets in the input universe of discourse $E \subset \Re^n$ to fuzzy sets in the output universe of discourse $V \subset \Re$ based on fuzzy logic principles. The fuzzy if-then rules are of the following form:

$$\mathcal{R}_l : \text{if } x_1 \text{ is } F_1^l \text{ AND } \cdots \text{ AND } x_n \text{ is } F_n^l \text{ THEN } y \text{ is } G^l. \quad (3.1)$$

where F_i^l and G^l are fuzzy sets, $\mathbf{x} = (x_1, \dots, x_n)^T \in E$ and $y \in V$ are input and output linguistic variables, respectively, and $l = 1, 2, \dots, M$.

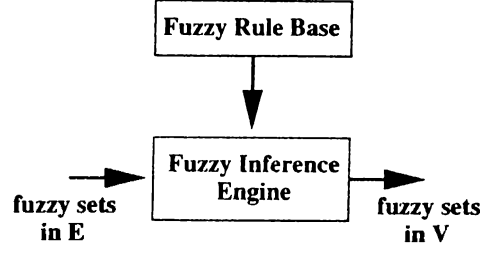


Figure 3.2: Basic configuration of pure FLS

3.2.2 Takagi and Sugeno's FLS

Instead of considering the fuzzy if-then rules in the form of Eqn.3.1, Takagi and Sugeno proposed to use the following fuzzy if-then rules:

$$\begin{aligned} \mathcal{R}_l : & \text{ if } x_1 \text{ is } F_1^l \text{ AND } \cdots \text{ AND } x_n \text{ is } F_n^l \\ & \text{ THEN } y^l = c_0^l + c_1^l \times x_1 + \cdots + c_n^l \times x_n. \end{aligned} \quad (3.2)$$

where F_i^l are fuzzy sets, c_i are real-valued parameters, y^l is the system output due to rule $L^{(l)}$, and $l = 1, 2, \dots, M$.

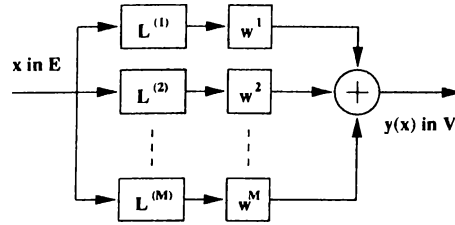


Figure 3.3: Basic configuration of Takagi and Sugeno's FLS

3.2.3 FLS with Fuzzifier and Defuzzifier

In order to use the pure FLS shown in Fig.3.2 in engineering systems where inputs and outputs are real-valued variables, the most straightforward way is to add a fuzzifier to the input and a defuzzifier to the output of the pure FLS.

The basic configuration of fuzzy logic system with fuzzifier and defuzzifier is shown in Fig.3.4 The fuzzifier maps real-valued inputs (crisp points) in E to fuzzy sets in E , and the defuzzifier maps the fuzzy sets in V to real-valued control outputs (crisp points) in V .

In this thesis while designing a fuzzy controller, the structure of the fuzzy logic systems with fuzzifier and defuzzifier are used. The computational structure of this type of system will be given in the next section. More detail for the other two types of fuzzy systems can be found in [9].

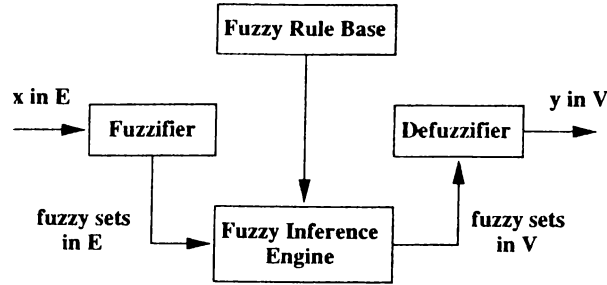


Figure 3.4: FLS with fuzzifier and defuzzifier

3.3 Computational Structure of FLS with Fuzzifier and Defuzzifier

The computational structure of a FLS consists of a number of computational steps as presented in Fig.3.5. There are five such computational steps.

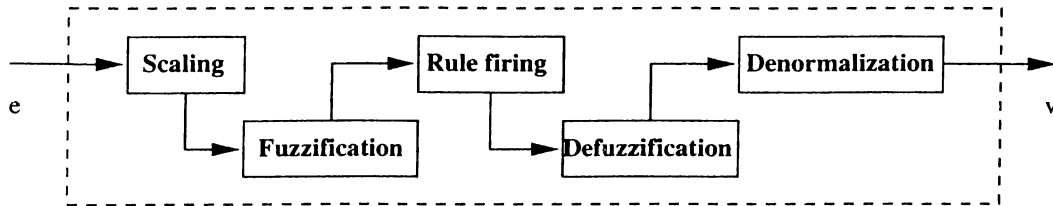


Figure 3.5: Computational Structure of FLS with fuzzifier and defuzzifier

A typical FLS has a number of rules. These rules for example can have the following form :

$$\mathcal{R}_s : \text{if “} e_1 \text{ is } LE_1^{(s)} \text{” AND ... AND “} e_m \text{ is } LE_m^{(s)} \text{” THEN “} v \text{ is } LV^{(s)} \text{”}. \quad (3.3)$$

where $LE_1^{(s)}, \dots, LE_m^{(s)}$ are the linguistic values taken by the process state variables (linguistic variables) e_1, \dots, e_m in the rule-antecedent of the s^{th} rule. The meaning of each linguistic value $LE_k^{(s)}$ is represented by a membership function defined on a domain, say, ϵ of the process variable e_k . Thus the meaning of $LE_k^{(s)}$ is given by $\mu_{LE_k^{(s)}} : \epsilon \rightarrow [0, 1]$. The objective is to find a value for the control variable v using a set of rules in the form of Eqn.3.3.

The states of a system can be measured with some sensors. These sensors provide, for example, some voltage values like 1.23V which is a crisp value.

The range of these different sensors might be different and therefore the domain of the membership functions related with the propositions in the rules will be different. If all of these membership functions are defined on a common domain then the inputs (state values) must be scaled so that each will be mapped onto its physical input domain. *Input Scaling* is not needed when the definitions of the membership functions are made on the physical domain of the inputs.

So far, the described fuzzy calculations are made on fuzzy domain, that is the arguments of the fuzzy operators are all fuzzy sets. Therefore to use the measurement of the sensors, one has to convert these crisp values into fuzzy set. This is called *fuzzification*. The result of fuzzy calculations are again fuzzy sets. But the controller output must be a crisp value to drive the system. Therefore a *defuzzification* module is needed to convert the fuzzy set to a suitable crisp value.

In section Sec.2.4 the calculations of the relation corresponding to an if-then statement with Mamdani implication is given. Performing this calculation one can obtain the relation related with the rule, and with the composition operation given in Sec.2.3, the fuzzy set corresponding to the controller output can be calculated.

These calculations can be performed for an FLS with a single rule in a straightforward way. For an FLS with multi-rule these calculations can be

performed in two different ways. One of them is the *composition based inference* and the other is the *individual-rule based inference*. These two methods result in the same fuzzy set for the controller output if Mamdani implication is used to represent the meaning of the rules. There exist some implication operations which result in different fuzzy sets for the two method of inference. One of these implications is Gödel implication[7].

In composition based inference, first all the relations for each rule are calculated. Then with a fuzzy union(OR) operator, these relations are combined into a single relation representing the complete rule base. Then the composition of the fuzzified input and this overall relation results in the fuzzy set for the controller output. In other words,

$$\begin{aligned} \mathcal{U} &= \mu^* \circ \mathcal{R} \quad \text{where, } \mu^* \text{ is the membership function after fuzzification of} \\ &\quad \text{the input variables, } \mathbf{e}^* = (e_1, \dots, e_m), \\ \mathcal{R} &= \bigcup_{s=1}^M \mathcal{R}_s, \\ \mathcal{R}_s &\text{ is the relations corresponding to the } s^{th} \text{ rule.} \end{aligned} \tag{3.4}$$

An if-then rule is a relation between the corresponding domain of the proposition in its antecedent and consequent parts. Rule firing is the process of obtaining the fuzzy set for the consequent proposition using the relation corresponding to the if-then rule and the observed crisp input values. Rule firing is given in more detail in Sec.3.3.2. In individual-rule based inference, each rule is fired individually and a fuzzy set for the controller output is obtained. Then the union of these fuzzy sets gives the resultant fuzzy set for the controller output. Composition based inference of rules requires much more calculations than individual-rule based inference, therefore the second inference method is used to get the fuzzy set for the controller output.

3.3.1 Fuzzification

Fuzzification operation depends on the inference method used. For composition based inference, fuzzification of the input, $\mathbf{e}^* = (e_1^*, \dots, e_m^*)$ means a fuzzy set

having the membership function μ^* which is defined as follows :

$$\mu^*(e_1, \dots, e_m) = \begin{cases} 1 & \text{for } e_1 = e_1^*, \dots, e_m = e_m^* \\ 0 & \text{otherwise.} \end{cases} \quad (3.5)$$

For the individual-rule based inference, the fuzzification operation means the following. The states of a system can be measured by some sensors. These sensors can provide, for example, some voltage values like 1.23V, which is a crisp value. In the rule-antecedent of the if-then statement there exists fuzzy propositions like “ e_k is $LE_k^{(s)}$ ”. It is known that $e_k^* = 1.23V$ and to perform fuzzy calculations related with the rules, it is required to have a truth value for the proposition “ e_k is $LE_k^{(s)}$ ”. For this method of inference *fuzzification* is the process of finding the membership degree of e_k^* in $\mu_{LE_k^{(s)}}$. And this is done for every element of \mathbf{e}^* where $\mathbf{e}^* = (e_1^*, \dots, e_m^*)$

Example : Consider the rules

R_1 : if “ e is *SMALL*” THEN “ v is *SMALL*”,

R_2 : if “ e is *BIG*” THEN “ v is *BIG*”.

In Fig.3.6 the related membership functions for e and v are given. Let $e^* = 2.5$. Then from Fig.3.6 we obtain the degrees of membership $\mu_{SMALL}(2.5) = 0.25$. So the fuzzification of $e^* = 2.5$ for the first rule, R_1 , means the membership value of $\mu_{SMALL}(2.5) = 0.25$. Similarly the fuzzification of $e^* = 2.5$ for the second rule, R_2 means the membership value of $\mu_{BIG}(2.5) = 0.75$

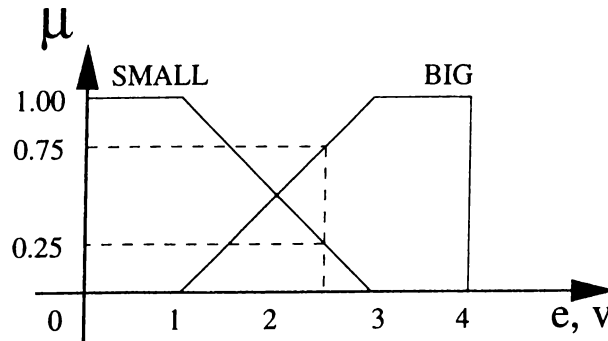


Figure 3.6: Membership functions

3.3.2 Rule Firing

Rule firing is to calculate a membership function for the proposition in the consequent part of the if-then statement. Let the s^{th} rule be as in Eqn.3.3. Then with the fuzzification operation the truth values for the propositions “ e_k is $LE_k^{(s)}$ ” is known. With a fuzzy *AND* operator the truth value of the antecedent part of the if-then statement can be calculated. The result of this calculation will be a crisp value since the inputs are crisp values. Next from Sec.2.4, one of the implication procedure is chosen. The fuzzy set for the control output is obtained by calculating the result of the implication operation for the calculated truth value of the antecedent part of the if-then statement with every control value. For Mamdani implication this fuzzy set will be calculated by the following equation.

$$\forall v \in V : \mu_{CLV}(v) = \min(\min(\mu_{LE_1^{(s)}}(e_1^*), \dots, \mu_{LE_m^{(s)}}(e_m^*)), \mu_{LV}(v)). \quad (3.6)$$

Example : Consider again the same if-then statements in the previous example. The propositions has a truth value 0.25 for the first rule and 0.75 for the second rule. If we chose Mamdani implication for the inference engine then the firing of a rule means to clip the membership function given in the consequent part of the rule at the point of the truth value of the antecedent part.

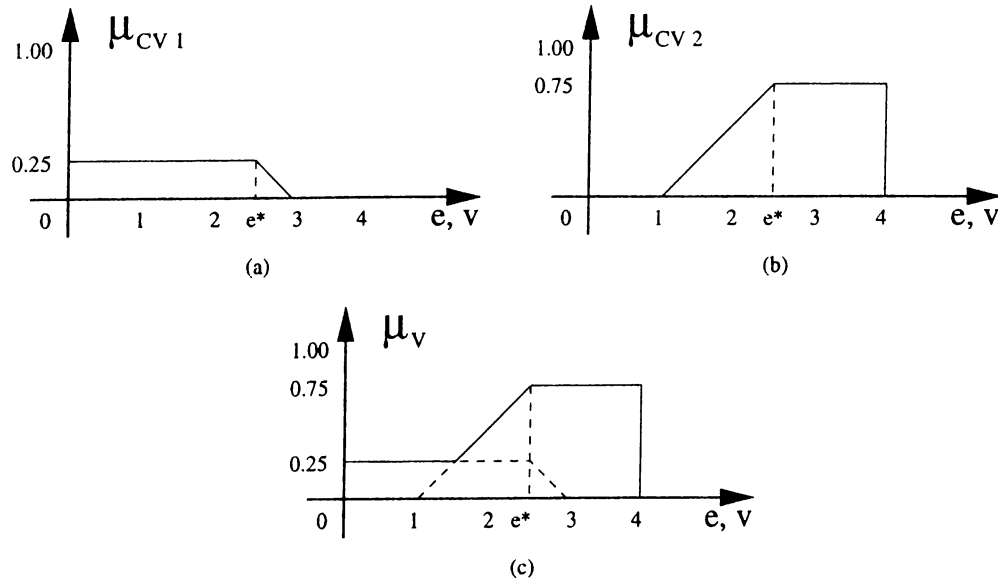


Figure 3.7: Clipped fuzzy sets as a result of rule firing

Therefore firing the first rule will result a membership function for v as given in Fig.3.7.a. For the second rule the membership function BIG will be clipped at point 0.75 as shown in Fig.3.7.b. To use all the information supplied by each one of the rules, we have to take the union of these clipped fuzzy sets. Choosing the union operator as the maximum operation we find the fuzzy set for the linguistic variable v as given in Fig.3.7.c. This is the fuzzy set to which the defuzzification operation must be applied.

3.3.3 Defuzzification

As explained previously the output of the inference process is a fuzzy set. In the on-line control, a non-fuzzy(crisp) control action is usually required. Consequently, one must defuzzify the control output inferred from the fuzzy control algorithm. There are many defuzzification methods proposed by many researchers[7],[8],[2],[9]. One of these method is center of gravity and used in the simulations of this thesis. This defuzzification method calculates the gravity point of the membership function μ_v with respect to the control variable v and sets the gravity center as the defuzzified control output.

3.4 Design of Fuzzy Logic Controller

The design parameters of a fuzzy logic controller(FLC) can be summarized as follows:

Inference Method :

- Choice of inference method

Fuzzification Module :

- Choice of fuzzification method

Rule Base :

- Choice of Variables and Content of Rules
- Choice of a Term Set
- Derivation of Rules

Data Base :

- Choice of Membership Functions
- Choice of Scaling Factors

Defuzzification Module :

- Choice of defuzzification method

After choosing the method for the inference one also has made the choice for the fuzzification operation as explained in the previous section. Choice for the input and control variable is done according to the problem in hand. Typically, the linguistic variables in a FLS are the state, state error, derivative, state error integral, etc. There is no well established method for the choice of term set, which is the linguistic values like *small*, *positive big*, *slightly increasing etc.*, membership function, scaling factors, and defuzzification method. Most of the time the choices are done by trial and error method. Therefore experience and engineering knowledge play an important role during this selection stage. There are some studies to determine these parameters from the input output data of the system [10], [11] and studies related with the effects of the variations of these parameters to the performance of the overall system [2], [7].

3.4.1 Derivation of Rules

There are five approaches for the derivation of fuzzy control rules.

Expert Experience and Control Engineering Knowledge: The formulation of fuzzy control rules can be achieved by means of two heuristic approaches. The most common one involves an introspective verbalization of human expertise. The other approach includes an interrogation of experts or operators using a carefully organized questionnaire.

Based on Operator's Control Actions: In many industrial man-machine control systems, the input-output relations are not known with sufficient precision to make it possible to employ classical control theory for modeling and simulation. And yet skilled human operators can control such systems quite successfully without having any quantitative models in mind. In effect, a human operator employs, consciously or subconsciously, a set of fuzzy if-then rules to control the process. In practice, such rules can be deduced from the observation of human controller's actions in terms of the input-output data.

Based on the Fuzzy Model of a Process: In linguistic approach, the linguistic description of the dynamic characteristics of a controlled process may be viewed as a fuzzy model of the process. Based on the fuzzy model, we can generate a set of fuzzy control rules for attaining optimal performance of a dynamic system.

Based on Learning: FLCs have been built to emulate human decision-making behavior, but few are focused on human learning, namely, the ability to create fuzzy control rules and to modify them based on experience. Procyk and Mamdani described the first self-organizing controller(SOC). The SOC has a hierarchical structure which consists of two rule bases. The first one is the general rule base of an FLC. The second one is constructed by "meta-rules" which exhibit human-like learning ability to create and modify the general rule base based on the desired overall performance of the system.

Based on Input-Output Data of the System: In this approach, the desired input to the system, which is the output of the controller, and the input to the controller, which is due to the corresponding system output must be known.

In [11], fuzzy clustering is considered as an intuitive approach of objective rule generation in fuzzy modeling. This method will be considered in more detail in the following chapters.

In this thesis, rules were derived based on engineering knowledge and, based on the input-output data of the system. The design of FLCs with the rules generated using these two approaches are discussed in the following two chapters.

Chapter 4

Fuzzy Logic Controller Design : Heuristic Rule Generation

In this chapter the development of a fuzzy logic controller will be investigated. Fuzzy rules and reasoning are utilized on-line to determine the controller parameters based on the error signal and its first derivative. The rule bases are constructed by examining a typical step response and by observing the effects of the variations of the controller parameters on the output of the system.

4.1 PID Controller Design

The best known controllers used in industrial control processes are proportional-integral-derivative(PID) controllers because of their simple structure and robust performance in a wide range of operating conditions. The design of such a controller requires specifications of three parameters: proportional gain(K_p), integral gain(K_i), and derivative gain(K_d).

The PID controllers in the literature can be divided into two main categories. In the first category, the controller parameters are fixed after they have

been tuned or chosen in a certain optimal way. The Ziegler-Nichols tuning formula is perhaps the most well-known tuning method. The controllers of the second category have a structure similar to PID controllers, but their parameters are adapted on-line based on parameter estimation, which requires certain knowledge of the process.

The system structure under consideration is given in Fig.4.1. Here $P(s)$ is the transfer function of the plant and the FLCs are the fuzzy logic systems tuning each parameter of the PID controller. The objective is to construct a rule base and to set the related membership functions for each FLC to get a satisfactory output response.

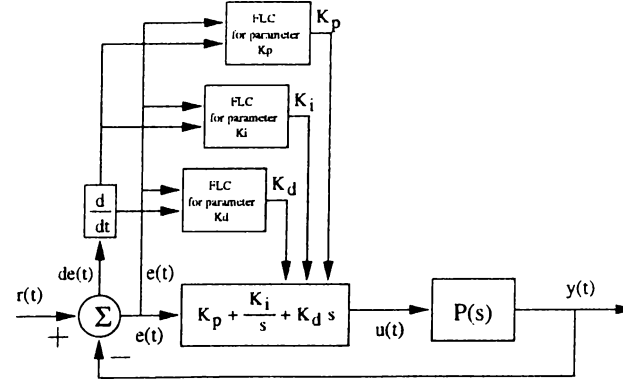


Figure 4.1: System structure under investigation

4.1.1 Rule-Base Construction

The fuzzy rules for the PID controller is derived from the examination of a typical step response given in Fig.4.2. In this figure the sign of error($e(t)$) and increment of the error($de(t)$) are given in a tabular form. The points where $e(t)$ and $de(t)$ changes their signs are labeled on the step response with the letters a, b, c, \dots, j . The following argument for the derivation of the rules will be similar to the work done by Tomizuka et.al.(1993)[4].

Consider the point **a** given on the step response in Fig.4.2. At the beginning a big control signal is required in order to achieve a fast rise time. To produce a big control signal, the PID controller should have a large proportional gain,

a large integral gain and a small derivative gain. From point **a** to **b** the error is positive and big. The increment of error is negative and the magnitude seems to be big in the middle of these points. For some slow responding systems at points closer to **a**, the magnitude of the increment of the error can be small. From the above discussion one may derive the following rules:

$$\begin{aligned}
 &\text{if "e(t) is PB" and "de(t) is NB" then "K}_p \text{ is BIG",} \\
 &\text{if "e(t) is PB" and "de(t) is NB" then "K}_i \text{ is BIG",} \\
 &\text{if "e(t) is PB" and "de(t) is NB" then "K}_d \text{ is SMALL".}
 \end{aligned} \tag{4.1}$$

$$\begin{aligned}
 &\text{if "e(t) is PB" and "de(t) is NS" then "K}_p \text{ is BIG",} \\
 &\text{if "e(t) is PB" and "de(t) is NS" then "K}_i \text{ is BIG",} \\
 &\text{if "e(t) is PB" and "de(t) is NS" then "K}_d \text{ is SMALL".}
 \end{aligned} \tag{4.2}$$

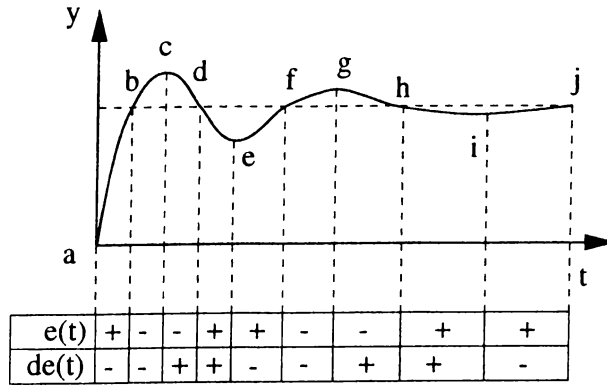


Figure 4.2: A typical step response of a system.

Here PB, NS stands for Positive Big and Negative Small which are fuzzy sets. As it is clear the first letter stands for the sign and the second for the magnitude. Fuzzy sets defined on the domain of error and increment of the error are labeled with the following: NB, NM, NS, ZE, PS, PM, PB. Here *M* stands for Medium, ZE for zero.

At points closer to **b** the error is close to zero but the increment of the error is negative and big. At this point to avoid a large overshoot the PID

controller should have a small proportional gain, a large derivative gain, and a small integral gain. Thus the following rules are given:

$$\begin{aligned}
 &\text{if "e(t) is ZE" and "de(t) is NB" then "K}_p \text{ is SMALL",} \\
 &\text{if "e(t) is ZE" and "de(t) is NB" then "K}_i \text{ is SMALL",} \quad (4.3) \\
 &\text{if "e(t) is ZE" and "de(t) is NB" then "K}_d \text{ is BIG".}
 \end{aligned}$$

By using similar arguments, a set of rules, as shown in Tab.4.1, may be used to adapt the proportional gain. The tuning rules for derivative gain and integral gain are given in Tab.4.2 and Tab.4.3, respectively.

| | | de(t) | | | | | | |
|------|----|-------|----|----|----|----|----|----|
| | | NB | NM | NS | ZE | PS | PM | PB |
| e(t) | NB | M | B | B | B | B | B | M |
| | NM | M | M | B | B | B | M | M |
| | NS | S | M | M | M | M | M | S |
| | ZE | S | S | S | S | S | S | S |
| | PS | S | M | M | M | M | M | S |
| | PM | M | M | B | B | B | M | M |
| | PB | M | B | B | B | B | B | M |

Table 4.1: Rules for proportional gain

| | | de(t) | | | | | | |
|------|----|-------|----|----|----|----|----|----|
| | | NB | NM | NS | ZE | PS | PM | PB |
| e(t) | NB | M | S | S | S | S | S | M |
| | NM | M | M | S | S | S | M | M |
| | NS | B | M | M | M | M | M | B |
| | ZE | B | B | M | M | M | B | B |
| | PS | B | M | M | M | M | M | B |
| | PM | M | M | S | S | S | M | M |
| | PB | M | S | S | S | S | S | M |

Table 4.2: Rules for derivative gain

In these tables, "S", "M", and "B" refer to "Small", "Medium", and "Big", respectively.

| | | de(t) | | | | | | |
|------|----|-------|----|----|----|----|----|----|
| | | NB | NM | NS | ZE | PS | PM | PB |
| e(t) | NB | M | B | B | B | B | B | M |
| | NM | M | M | B | B | B | M | M |
| | NS | S | M | M | M | M | M | S |
| | ZE | S | S | M | M | M | S | S |
| | PS | S | M | M | M | M | M | S |
| | PM | M | M | B | B | B | M | M |
| | PB | M | B | B | B | B | B | M |

Table 4.3: Rules for integral gain

4.1.2 FLC Structure

The membership functions for the linguistic variables error and increment of error are given in Fig.4.3. All the membership functions of error and its increment are defined on the interval $[0,1]$. The actual domain of the linguistic variable error is the unit interval but the domain of its increment depends on the simulation step size and its variation is much smaller than the unit interval. Therefore a normalization procedure is applied to the increment of the error before the fuzzification operation. The scaling factor in this normalization operation is chosen so that the increment of error is large enough to span the interval $[-1, 1]$.

In designing the controller *individual rule-based inference* is chosen. Therefore the fuzzification operation is performed as explained in the example of Sec.3.3.1. The fuzzy set operations used in the design are the same as given in Def.3. Mamdani implication is used in firing each rule.

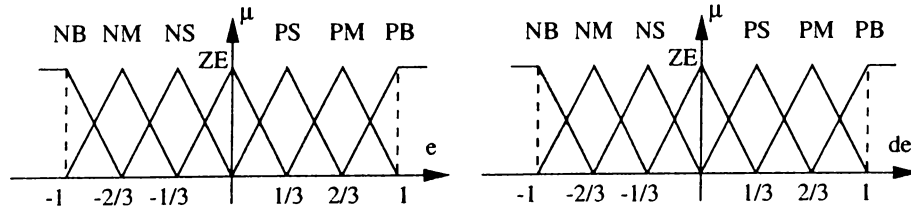


Figure 4.3: Membership functions for error and increment of error

After firing each rule in the rule-base of a parameter there will be clipped version of the fuzzy sets given in Fig.4.4 for the linguistic variables K_p , K_i ,

and K_d . The gains of the PID controller are always positive. Therefore before the scaling of the domain of these membership functions they must be shifted to a reasonable amount.

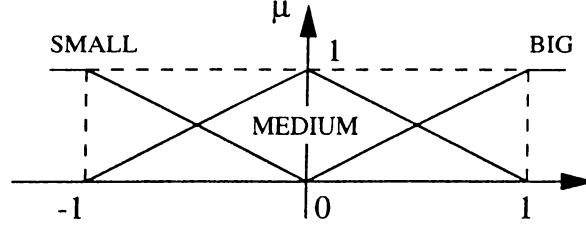


Figure 4.4: Membership function for the variables K_p , K_i , and K_d

In general the determination of this shift and the scaling is a trial and error procedure. Based on extensive simulation study on various processes, a rule of thumb for determining the range of K_p and the range of K_d is given as

$$K_{p,min} = 0.32K_{cr}, \quad K_{p,max} = 0.6K_{cr},$$

$$K_{d,min} = 0.08K_{cr}P_{cr}, \quad K_{d,max} = 0.15K_{cr}P_{cr}.$$

where K_{cr} and P_{cr} are the gain and the period of oscillation at the stability limit under P-control[12]. Tomizuka et.al.[4] uses this method to set the domain of the membership functions for the gains because in their work they related the integral gain to the proportional and derivative gain with a constant α . In our design all PID gains are considered to be independent of the each other and each of them have a separate rule base. In the simulations the ranges of the membership functions are found by first applying this method and then making a fine tuning.

4.1.3 Simulations and Results

The fuzzy parameter tuning has been tested on a variety of processes. One of them is given here. The plant transfer function is given in Eqn.4.4. Ziegler-Nichols PID tuning results in the following gains for this plant.

$$P_1(s) = \frac{1}{s^3 + 6s^2 + 11s + 6}, \quad (4.4)$$

$$K_p = 36.0000, \quad K_i = 38.0047, \quad K_d = 8.5253.$$

The membership functions for error and change of error is the same as given in Fig.4.3. The normalization constant for the increment of the error was chosen to be 20.

The membership functions for K_p , K_i , and K_d are given in Fig.4.5. These figures show the effective membership functions taking into account the shift and scaling operations in the denormalization stage.

Computation of the parameters K_p , K_i , and K_d

Assume that the error and its increment is \mathbf{e}^* and \mathbf{de}^* , respectively. Let us take one of the entries of Tab.4.1.

$$\text{if "e(t) is PS" and "de(t) is NS" then } \sim K_p \text{ is MEDIUM".} \quad (4.5)$$

First the fuzzification operation must be applied to the crisp values \mathbf{e}^* and \mathbf{de}^* . Since individual-rule based inference is chosen in the calculations, the fuzzification operation will result the following two value, $\mu_{PS}(\mathbf{e}^*)$ and $\mu_{NS}(\mathbf{de}^*)$.

The set operations were chosen as the min, max, and standard complement. So the truth value of this rule will be calculated as :

$$\mu_i = \min(\mu_{PS}(\mathbf{e}^*), \mu_{NS}(\mathbf{de}^*)). \quad (4.6)$$

For the implication operation, Mamdani implication is chosen. Therefore the implied fuzzy set for the proportional gain is calculated as,

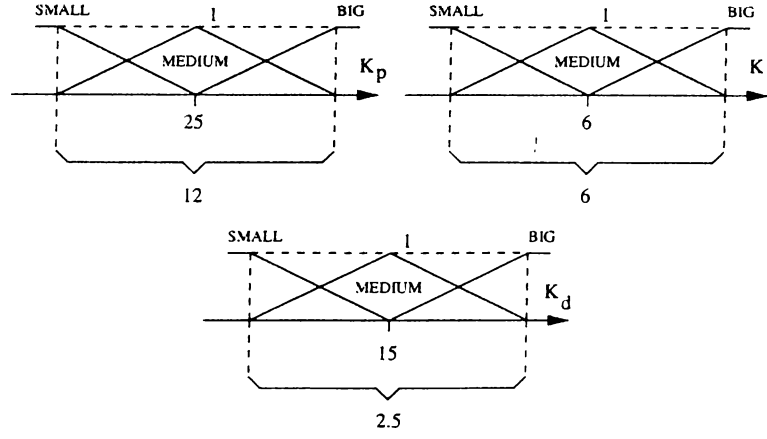


Figure 4.5: Membership function for the variables K_p , K_i , and K_d

$$\mu_{CM}^i(u) = \min(\mu_i, \mu_M(u)). \quad (4.7)$$

This procedure is performed to all of the rules in the rule table and all of these clipped fuzzy sets are combined with the union operator, $\max(\cdot)$.

$$\mu(u) = \max_u(\mu_M(u)^1, \mu_M(u)^2, \dots, \mu_M(u)^N). \quad (4.8)$$

This way the final fuzzy set for the parameter is obtained. This value must be converted to a crisp value by the defuzzification operation. In this thesis the center of gravity is used as the defuzzification operation. Applying this to the fuzzy set $\mu(u)$ gives the value for the proportional gain.

$$K_p^* = \frac{\sum_{i=1}^l u_i \mu(u_i)}{\sum_{i=1}^l \mu(u_i)}. \quad (4.9)$$

This operation is repeated for K_i and K_d in the same way. The output of the system with Ziegler-Nichols PID controller and the output of the system where the parameters are tuned with FLCs are compared in Fig.4.6. The PID parameters calculated by the FLCs are given in Fig.4.7

The designed controller is tested for a plant $P_2(s)$ given below, which may be considered as a slightly perturbed version of the plant given by Eqn.4.4.

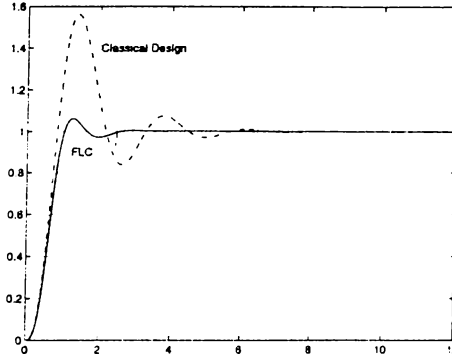


Figure 4.6: Comparison of the step responses.

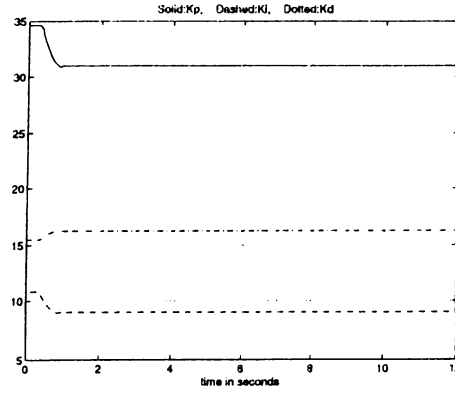


Figure 4.7: PID parameters determined by the FLCs.

$$P_2(s) = \frac{2}{s^3 + 4s^2 + 11s + 6}. \quad (4.10)$$

The output waveforms for the fuzzy PID and the classical PID are compared in Fig.4.8. The PID parameter variations are given in Fig.4.9.

The proposed gain scheduling scheme uses fuzzy rules and reasoning to determine the PID parameters. This fuzzy PID controller design has been tested on various processes in simulation, and satisfactory results are obtained. From Fig.4.9 it is seen that with the proposed PID controller the overall system is at least as insensitive to the parameter variations as the system with PID controller, having fixed gains.

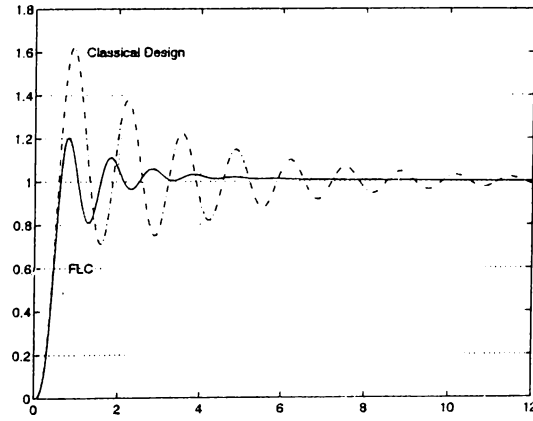


Figure 4.8: Comparison of the step responses.

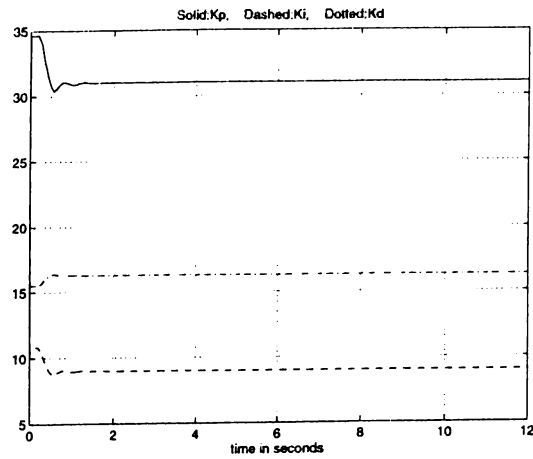


Figure 4.9: PID parameters determined by the FLCs.

4.2 Lead-Lag Controller Design

For this design problem the structure of the overall system is as shown in Fig.4.10. In this figure $P(s)$ is the plant transfer function. There are two parameters in the lead-lag controller to be determined, namely a and T .

As in the case of PID controller we need two FLC blocks, one for a and one for T . The effect of the variations of the parameters a and T on the output of the system seems to be in the same direction. Therefore the parameter T is fixed and only a is tuned with FLC. The value of T is determined with a classical design method.

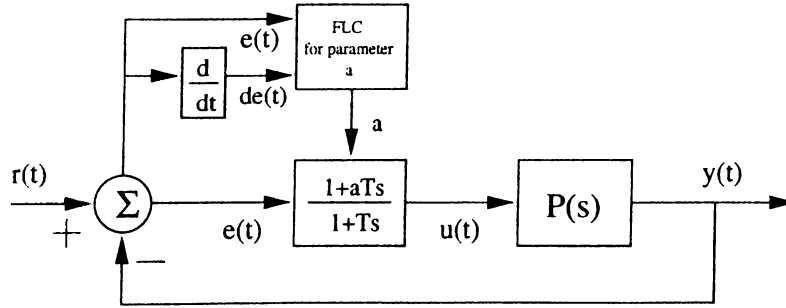


Figure 4.10: System structure under investigation

4.2.1 Rule-Base Construction

To determine the effect of the parameters on the step response of the system, many simulation have been performed. In all of the cases the effect of a and T turn out to be the same. Therefore only the parameter a will be investigated. The reason to choose the parameter a for the tuning is that, for values of a less than 1 the controller is a phase lag controller and for a greater than 1 it is a phase lead controller. The value of T is set to 1.

In Fig.4.11 and Fig.4.12 the variations of the step responses of two different plants, given above with respect to the parameter a are shown. In these figures the solid lines are the output of the plants $P_1(s)$ and $P_2(s)$ without a controller.

These plants have the following transfer functions:

$$P_1(s) = \frac{27}{s^2 + 10s + 36}, \quad P_2(s) = \frac{72}{s^3 + 10s^2 + 36s}.$$

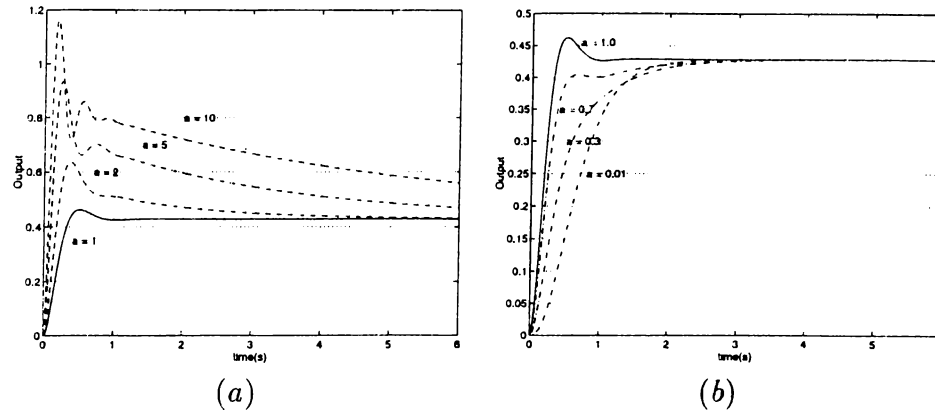


Figure 4.11: Step responses of $P_1(s)$ for $a=0.01, 0.3, 0.7, 1, 2, 5, 10$.

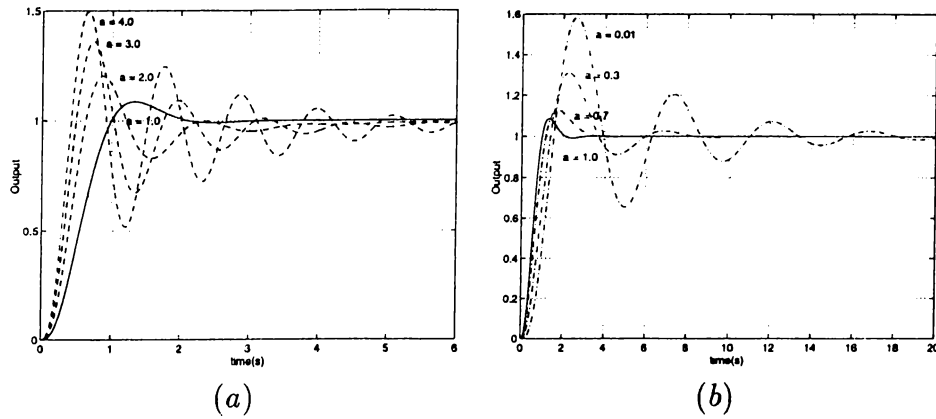


Figure 4.12: Step responses of $P_2(s)$ for $a=0.01, 0.3, 0.7, 1, 2, 3, 4$.

In Fig.4.11.(a) the step responses for $a = 1, 2, 5, 10$ are given. As the value of a increases the rise time decreases and the overshoot increases. In Fig.4.11.(b) the step responses for $a = 0.01, 0.5, 0.7, 1$ are shown. Here the

same behavior is observed. The overshoot of the step response is proportional to the parameter a whereas there is an inverse relationship between the rise time of the system and the parameter a . In Fig.4.12.(b) the step responses for $a = 1, 2, 3, 4$ are given. As the value of a increases the rise time decreases and the overshoot increases. In Fig.4.12.(b) the step responses for $a = 0.01, 0.5, 0.7, 1$ are shown. Here a different behavior is observed. In this case both the overshoot of the step response and the rise time of the system are inversely proportional to the parameter a .

For the output of a system it is desired to have a short rise time and no overshoot and oscillations. To have a short rise time we have to chose a large value for a , say a_{max} . But for large a we get a large overshoot and rather an oscillatory response at the output. One possible way is to set the value of a to a large value in the beginning and reduce it when the system output is about to catch the reference signal. From Fig.4.2 at the beginning the error will be a member of the fuzzy set PB and its increment will be a member of the fuzzy set NB (or NS). So using these arguments one may state the following rules:

$$\begin{aligned} &\text{if "e(t) is PB" and "de(t) is NB" then "a is LA7",} \\ &\text{if "e(t) is PB" and "de(t) is NS" then "a is LA7",} \\ &\text{if "e(t) is ZE" and "de(t) is NB" then "a is LA6".} \end{aligned} \quad (4.11)$$

The meaning of the linguistic values LA7, LA6, and the rest are given in Fig.4.13. Using this discussion on the effect of the variation of a to the output a rule-base is constructed as given in Tab.4.4.

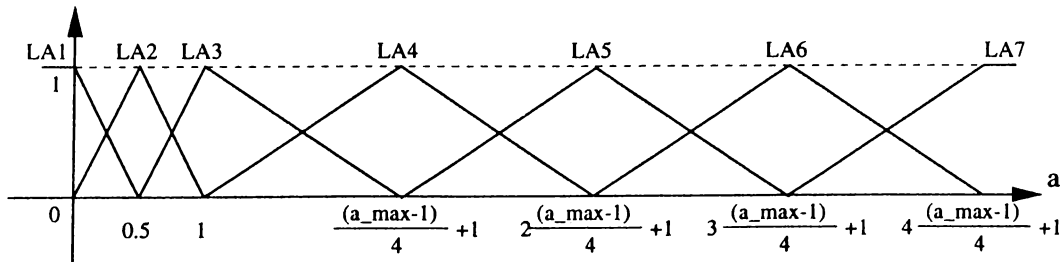


Figure 4.13: Membership functions of the linguistic variable a

The membership functions for the parameter a are chosen so that they all

| | | de(t) | | | | | | |
|------|----|-------|-----|-----|-----|-----|-----|-----|
| | | NB | NM | NS | ZE | PS | PM | PB |
| e(t) | NB | LA3 | LA3 | LA2 | LA2 | LA1 | LA1 | LA1 |
| | NM | LA4 | LA3 | LA3 | LA2 | LA2 | LA1 | LA1 |
| | NS | LA4 | LA4 | LA3 | LA3 | LA2 | LA2 | LA1 |
| | ZE | LA5 | LA4 | LA4 | LA3 | LA3 | LA2 | LA2 |
| | PS | LA6 | LA5 | LA4 | LA4 | LA3 | LA3 | LA2 |
| | PM | LA7 | LA6 | LA5 | LA4 | LA4 | LA3 | LA3 |
| | PB | LA7 | LA7 | LA6 | LA5 | LA4 | LA4 | LA3 |

Table 4.4: Rules for lead-lag parameter a

equally share the domain in the intervals $[0,1]$ and $[1, a_{max}]$.

4.2.2 FLC Structure

In Fig.4.13 the value of a_{max} is chosen to be a large number to have a short rise time. The membership functions for the error and its increments are as in the case of PID controller design. Again a normalization factor is used before the fuzzification operation such that the increment of error spans the interval $[-1, 1]$.

Individual rule-based inference is used to fire all the rules in the rule-base. Therefore the fuzzification operation is the same as in the previous section. The fuzzy set operations used in the design are the same as given in Def.3. As in the previous design here also Mamdani implication is used as the inference engine. Since the membership functions for the parameter a are defined on its actual domain there is no need for a denormalization stage.

4.2.3 Simulations and Results

This fuzzy parameter tuning scheme for the lead-lag type controller is applied to various systems. Simulations showed that a variety of processes can be satisfactorily controlled by the fuzzy parameter tuned lead-lag controller. One of the simulations is performed with a plant whose transfer function is given

in Eqn.4.12 . A phase-lead controller based on a classical design for this plant results in a controller whose transfer function is given in Eqn.4.13

$$P_3(s) = \frac{2500}{s(s + 25)}, \quad (4.12)$$

$$C(s) = \frac{0.016s + 1}{0.002s + 1}. \quad (4.13)$$

In this design the parameter a and T are 8 and 0.002 respectively[13]. In the proposed fuzzy parameter tuning algorithm the value of T is not changed. It is set to a value obtained in a classical design. Therefore for this plant the value of T is set to 0.002.

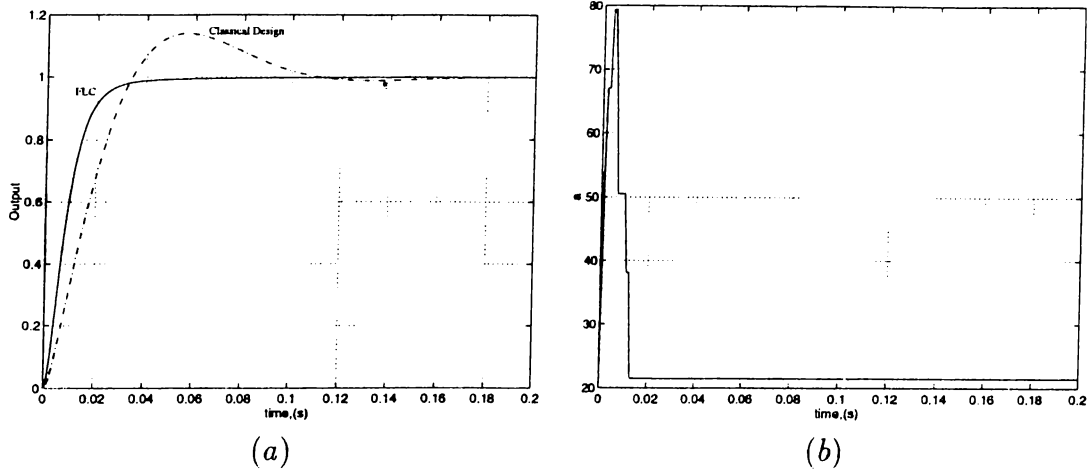


Figure 4.14: Comparison of the step responses of $P_3(s)$ with the two different controller.

The normalization constant for the increment of error is chosen to be 9. The membership function for the error and its increment is as mentioned in Sec.4.2.2. The membership functions for the linguistic values of a are taken as given in Fig.4.13. For this simulation a_{max} is set to 100. Again Mamdani Implication is used in interpreting the meaning of the rules.

The step responses of the system $P_3(s)$ with the two different controllers given above, one classical and one fuzzy, are given in Fig.4.14.(a). In

Fig.4.14.(b) the variation of the parameter a is given. From the figure it is seen that at the beginning a takes large values and as the output approaches the desired reference signal the FLC reduces the parameter a . This FLC has been tested on various systems which may be considered to be the plant $P_3(s)$ with distorted coefficients. One of these plants is given below:

$$P_4(s) = \frac{5000}{s^2 + 24s + 1}.$$

The step responses corresponding to the fuzzy tuned lead-lag controller and the classical design is given in Fig.4.15.(a)

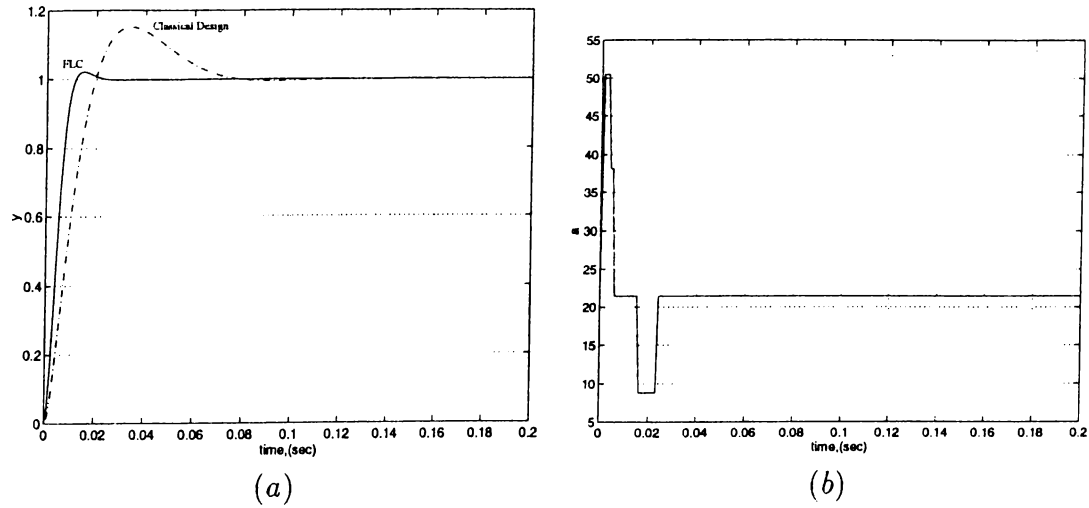


Figure 4.15: (a) Step responses of $P_4(s)$. (b) Variation of the parameter a .

In both of the designs the rise time reduces while a small increase in the overshoot of the system is observed. From the figure it is seen that even with the disturbed system the fuzzy tuned lead-lag controller performs better than the classical design.

Chapter 5

Fuzzy Controller Design : Rule generation using fuzzy c-means clustering algorithm

An if-then statement is a relation between the domains of its antecedent and consequent parts. Consider a fuzzy logic controller having a rule base whose if-then statements have only the error, e , in its antecedent part and the control variable, u , in its consequent part. Then the i^{th} if-then rule would be as the following type,

$$\text{if "e is } LE_i \text{" then "u is } LU_i \text{".} \quad (5.1)$$

where e and u are the linguistic variables for the error signal and the control signal, and LE_i and LU_i are the corresponding linguistic values. Let the membership functions of LE_i and LU_i be as given in Fig.5.1

In this figure the relation shown on the right hand side is obtained using Mamdani Implication. The value of the relation which is in the unit interval is the truth value of the if-then statement. As it is seen from the figure the closer the points to the centers of the membership functions, the closer a value

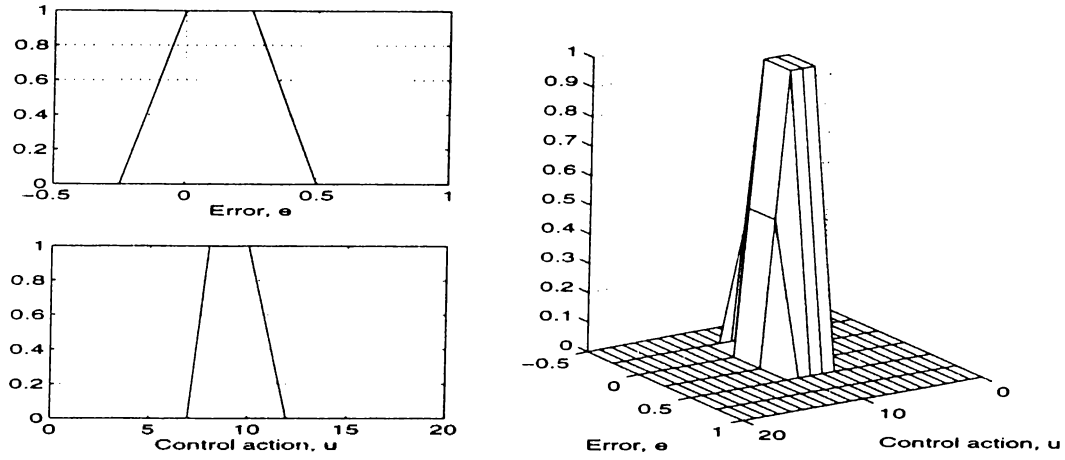


Figure 5.1: The membership functions and the relation of the rule if “ e is LE_i ” then “ u is LU_i ”.

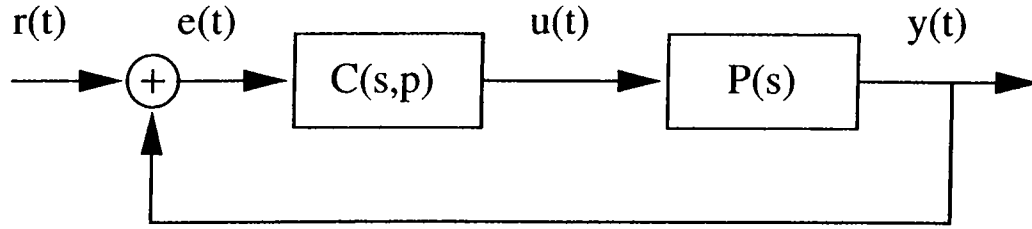


Figure 5.2: Negative unit feedback with a parametric controller $C(s,p)$

to 1 the relation assign to those points. Also one important thing is that the projection of this relation onto its subsets, error and control action, gives the membership functions for the linguistic values LE_i and LU_i .

Consider the system with the negative unit feedback controller given in Fig.5.2. $C(s,p)$ is the transfer function of a controller with a parameter vector p , and $P(s)$ is the transfer function of the plant. Assume that for this system, we know the control signal $u(t)$ which minimizes the desired cost function. Applying this control signal we will get the output signal, $y(t)$. From this output signal and the reference signal $r(t)$, we can calculate the error signal. Let the plot given in Fig.5.3 be obtained by plotting the control signal and the corresponding error signal.

It seen from Fig.5.3 that these data points can be clustered into three groups. The classical clustering algorithm forms the cluster boundaries so that

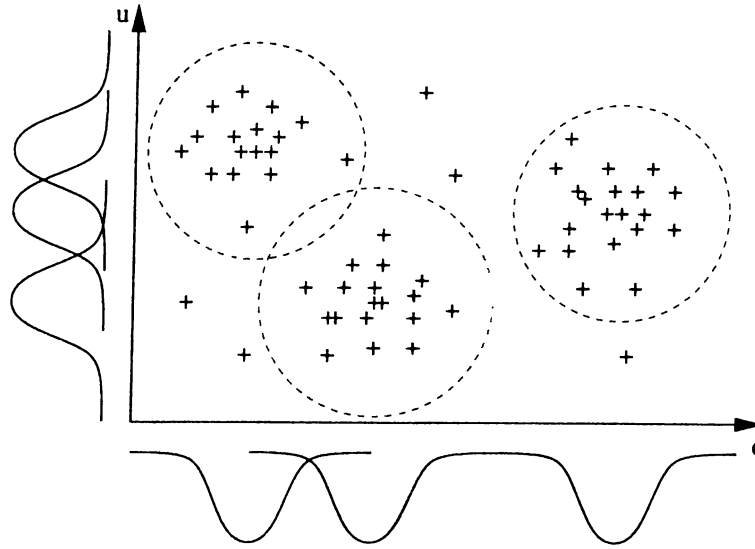


Figure 5.3: Sample point of error versus the corresponding control signal.

any sample point much closer to one cluster center belongs to that cluster. Therefore the sets the classical clustering algorithm defines for each cluster will have a characteristic function assigning 1 to those sample points which are closer to that cluster center and 0 otherwise. Fuzzy c-means clustering algorithm forms similar sets for each cluster but the characteristic function of these sets assigns values from the interval $[0,1]$. These characteristic functions assign values so that the closer the point to the center of the cluster the closer a value to 1 is assigned. Therefore the result of fuzzy c-means algorithm is a fuzzy relation between the error and the control action. If we consider Fig.5.1 then it can be concluded that each of these clusters, which the fuzzy c-means clustering algorithm calculates, is actually a relation corresponding to a rule in the fuzzy logic controller. The membership functions related with these rules are obtained by projecting these relations onto their subset.

In this chapter fuzzy c-means clustering algorithm is used to obtain the rules of a fuzzy logic controller which tunes the parameter of a controller. As it is noted, to use this algorithm it is necessary to know the optimal input to the system. The closed loop transfer function of the system given in Fig.5.2 is calculated to be :

$$G(s) = \frac{Y(s)}{R(s)} = \frac{C(s, \mathbf{p})P(s)}{1 + C(s, \mathbf{p})P(s)}. \quad (5.2)$$

Since the reference signal is known to be the unit step for our case, the actual inputs to the overall closed loop system are the parameters of the controller. Therefore the system whose optimal input must be calculated turns out to be a nonlinear one. To calculate the optimal control signal for a system of degree n there are $2n$ differential equation to solve. In our case these differential equations are nonlinear and most of the time the analytical solution of these equations can not be calculated. In some cases the numerical solutions encounter singularities in the integration procedure and may not give a solution. One other way to obtain a control action for the system is to use dynamic programming. Of course the solution of dynamic programming may not be the optimal solution because the state space and the input space are discretized but most of the time it is possible to obtain a solution to the problem.

5.1 Dynamic Programming

Consider a system described by the state difference equation

$$\mathbf{x}(k+1) = \mathbf{a}(\mathbf{x}(k), \mathbf{u}(k)) \quad k = 0, 1, \dots, N-1. \quad (5.3)$$

It is desired to determine the control law that minimizes the criterion

$$J = h(\mathbf{x}(N)) + \sum_{k=0}^{N-1} g_D(\mathbf{x}(k), \mathbf{u}(k)). \quad (5.4)$$

The application of dynamic programming to this problem leads to the recurrence equation

$$\begin{aligned}
J_{N-k,N}^*(\mathbf{x}(N-k)) &= \min_{\mathbf{u}(N-k)} \{g_D(\mathbf{x}(N-k), \mathbf{u}(N-k)) \\
&+ J_{N-(k-1),N}(\mathbf{a}_D(\mathbf{x}(N-k), \mathbf{u}(N-k)))\} \quad (5.5) \\
k &= 1, 2, \dots, N.
\end{aligned}$$

The solution of this recurrence equation is an optimal control law or optimal policy, $\mathbf{u}^*(\mathbf{x}(N-k), N-k), k = 1, 2, \dots, N$, which is obtained by trying all admissible control values at each admissible state values. To make the computational procedure feasible it is necessary to quantize the admissible state and control values into a finite number of levels [14].

5.2 Fuzzy C-Means Clustering Algorithm

The idea of fuzzy clustering is to divide the output data into fuzzy partitions which overlap with each other. Therefore, the containment of each data to each cluster is defined by a membership grade in $[0,1]$. In formal words, clustering in unlabeled data $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \subset \mathbb{R}^h$, where N is the number of data vectors and h is the dimension of each data vector, is the assignment of c number of partition labels to the vectors in X . A c -Partition of X is a set of $(c \cdot N)$ membership values $\{u_{ik}\}$ that can be conveniently arrayed as a $(c \times N)$ matrix $U = [u_{ik}]$. The problem of fuzzy clustering is to find the optimum membership matrix U . The most widely used objective function for fuzzy clustering in X is the weighted within-groups sum of squared error objective function J_m which is used to define the following constrained optimization problem.[10]

$$\min_{(U,V)} \left\{ J_m(U, V; X) = \sum_{k=1}^N \sum_{i=1}^c (u_{ik})^m \|\mathbf{x}_k - \mathbf{v}_i\|_A^2 \right\}, \quad (5.6)$$

where

$$U \in M_{fcn} = \left\{ U \in \mathbb{R}^{cN} \left| \begin{array}{l} 0 \leq u_{ik} \leq 1, \forall i, k \text{ and } \forall k, u_{ik} > 0 \exists i \\ 0 < \sum_{k=1}^N u_{ik} < N \forall i \text{ and } \sum_{i=1}^c u_{ik} = 1 \forall k \end{array} \right. \right\}. \quad (5.7)$$

$\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_c\}$ is a vector of (unknown) cluster centers, and $\|\mathbf{x}\|_A = \sqrt{\mathbf{x}^T \mathbf{A} \mathbf{x}}$ is any inner product norm. \mathbf{A} is a $h \times h$ positive definite matrix which specifies the shape of the clusters. We chose the matrix \mathbf{A} equal to the identity matrix which leads to the definition of Euclidean distance, and consequently to spherical clusters. Fuzzy partitions are carried out by the Fuzzy C-Means algorithm through an iterative optimization of Eqn.5.6 and Eqn.5.7 according to the following steps[10][15]

STEP 1 : CHOOSE

number of clusters (c), weighting exponent (m), iteration limit(iter)
 termination criterion ($\epsilon > 0$), norm for $J_m(\|\mathbf{x}_k - \mathbf{v}_i\|_A)$
 norm for error $\|\mathbf{V}_t - \mathbf{V}_{t-1}\|$.

STEP 2 : GUESS

initial position of cluster centers : $\mathbf{V}_0 = \{\mathbf{v}_{1,0}, \mathbf{v}_{2,0}, \dots, \mathbf{v}_{c,0}\} \subset \mathcal{R}^{\text{ch}}$

STEP 3 : ITERATION

FOR t = 1 to iter

$$u_{ik,t} = \left[\sum_{j=1}^c \left(\frac{\|\mathbf{x}_k - \mathbf{v}_{j,t-1}\|_A}{\|\mathbf{x}_k - \mathbf{v}_{i,t-1}\|_A} \right)^{\frac{2}{m-1}} \right]^{-1}$$

$$\mathbf{v}_{i,t} = \frac{\sum_{k=1}^N (u_{ik,t})^m \mathbf{x}_k}{\sum_{k=1}^N (u_{ik,t})^m}$$

IF error = $\|\mathbf{V}_t - \mathbf{V}_{t-1}\| \leq \epsilon$, THEN stop.

NEXT t

□

The fuzzy c-means clustering algorithm suffers from three major difficulties which are usually treated through heuristics for a specific problem at hand. These are the need to make a choice for the number of clusters(c), and weighting exponent(m), and that the different initial cluster centers will result in different cluster centers. In the simulations for the initial cluster centers, first a classical clustering algorithm is run over the set and the result of this clustering is used as the initial cluster centers for the fuzzy c-means algorithm. As a rule of thumb the value of m is chosen between 1.5 and 2.5. In our simulations this

value is set to 1.7. The number of clusters, c , is determined after running for different cluster numbers and choosing the best output for the system.

5.3 Formation of Membership Functions

The algorithm given in the previous section gives the membership value of each sample to each of the classes. To complete our rule base we need to define the membership functions of the linguistic values in the antecedent and consequent part of the if-then rules. A fuzzy version of the probabilistic classification method, which is called K-nearest neighbor has been introduced by Keller et al.[10] [16] and, is summarized in the following algorithm.

STEP 1 : DRIVE

$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\} \subset \mathbb{R}^h$ of labeled
data set with membership grade $\mathbf{U} = [u_{ij}]$.

For any \mathbf{x} of unknown classification,

STEP 2 : CHOOSE

number of neighbors (k) : $1 \leq k \leq n$; norm for the distance $\|\mathbf{x} - \mathbf{x}_j\|$

STEP 3 : LOOP

FOR $i = 1$ to c

Calculate the membership grade to \mathbf{x} among $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$

$$u_i(\mathbf{x}) = \frac{\sum_{j=1}^k u_{ij} \left(\|\mathbf{x} - \mathbf{x}_j\|^{\frac{2}{m-1}} \right)^{-1}}{\sum_{j=1}^k \left(\|\mathbf{x} - \mathbf{x}_j\|^{\frac{2}{m-1}} \right)^{-1}}$$

NEXT i

□

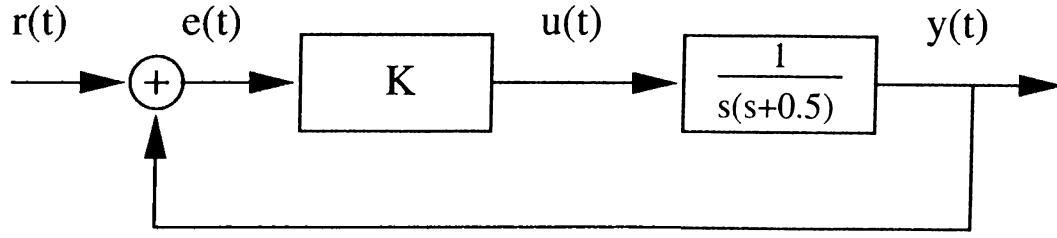


Figure 5.4: Negative unit feedback with a proportional controller K

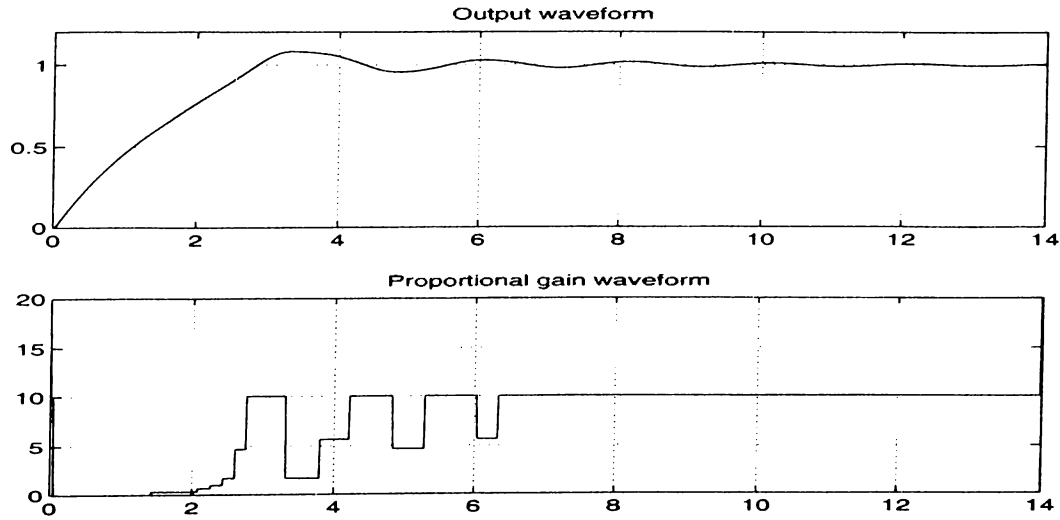


Figure 5.5: The suboptimum solution for the proportional control and the corresponding step response.

5.4 Simulations and Results

This algorithm is applied to a system whose transfer function is given with the unit feedback configuration in Fig.5.4

In this example the controller is taken to be a gain. For fuzzy c-means clustering algorithm we have to know the optimal waveform for K . Using dynamic programming the waveform is found to be as it is given in Fig.5.5. Since the state spaces and the input spaces are discretized in dynamic programming this waveform is suboptimal. In this figure the step response is the waveform at the output, $y(t)$, when the value of K is changed as it is calculated with dynamic programming.

Consider a system given by the following transfer function.

$$G(s) = \frac{Y(s)}{U(s)} = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}. \quad (5.8)$$

This system has a maximum overshoot of M_p which is given as

$$M_p = \exp^{-\left(\frac{\xi}{\sqrt{1-\xi^2}}\right)\pi}. \quad (5.9)$$

Solving this equation for K gives the following equation,

$$K = \frac{\left[1 + \left(\frac{\pi}{\ln M_p}\right)^2\right]}{16}. \quad (5.10)$$

For a maximum overshoot of $M_p = 0.2$ the value of K is obtained as 0.3008. Using the waveform given in Fig.5.5 for the K value and the corresponding error and its increment the rule base given in Fig.5.6 is obtained. The data vectors, which is used in fuzzy c-means clustering algorithm are obtained by augmenting the values of K , error, and the increment in a vector. So the data vectors are in \mathbb{R}^3 .

Various values are tried for the number of cluster, c . Each resulted in a different rule base. Among these rule bases the most effective one is chosen. This rule base contains just 7 rules. The output of the system with constant gain and fuzzy tuned gain are compared in Fig.5.7.(a). It is seen that the step response with fuzzy gain scheduling has less overshoot small rise time but high rate of oscillation. The waveform for K produced by the fuzzy logic controller is given in Fig.5.7.(b).

If this waveform is compared with the waveform calculated with dynamic programming, given in Fig.5.5, the similarity between the waveforms shows the ability of fuzzy c-means clustering algorithm in modeling a system. Therefore fuzzy c-means algorithm can also be used in system identification problem.

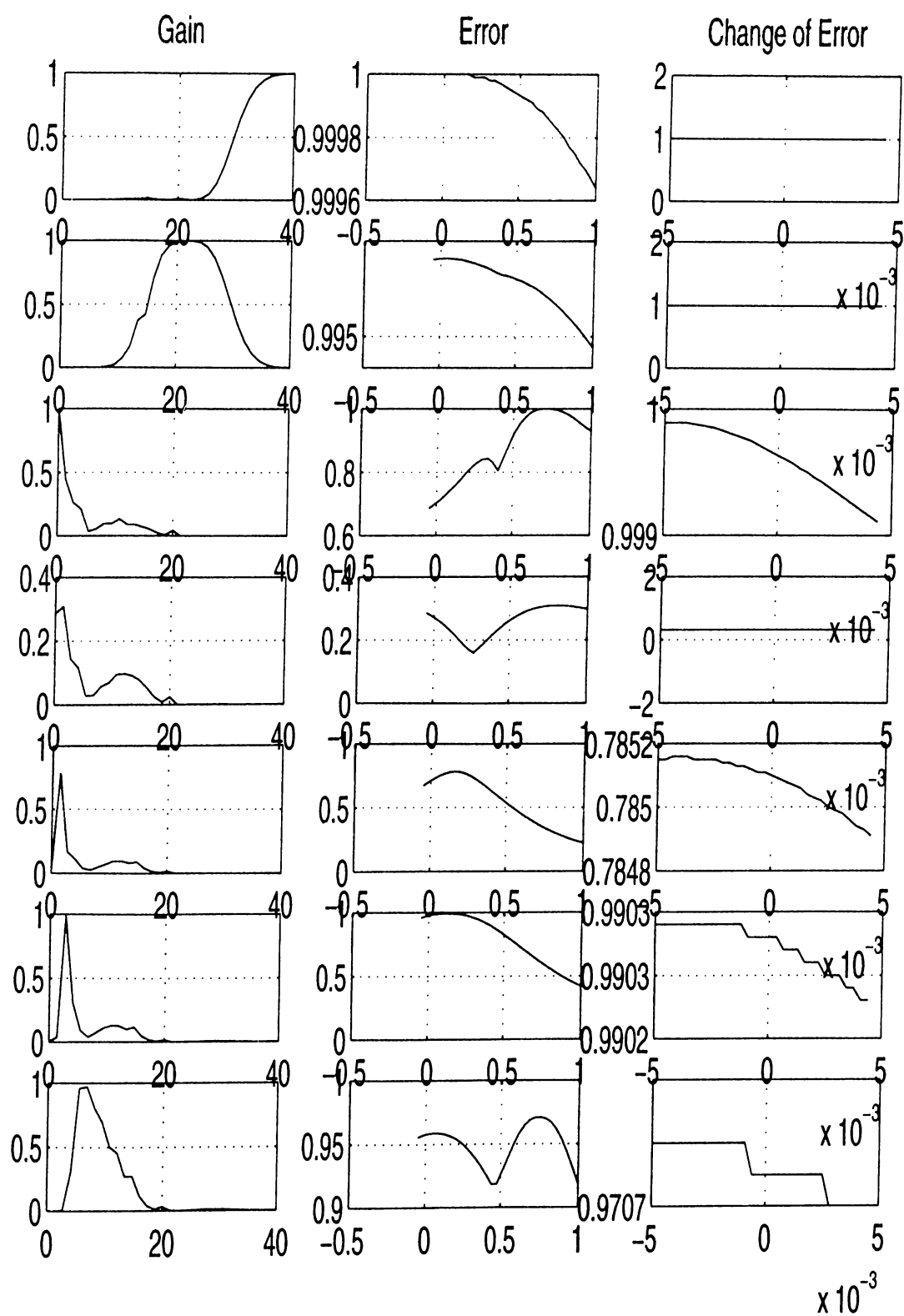


Figure 5.6: Rule base of the fuzzy controller

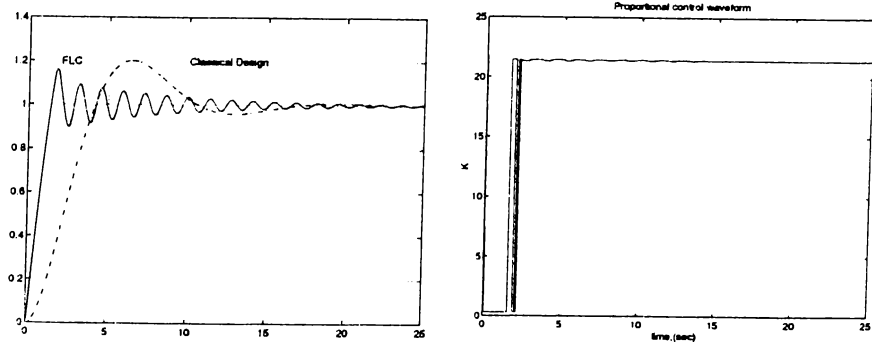


Figure 5.7: (a) The step responses with two different controller (b) The waveform for the proportional gain.

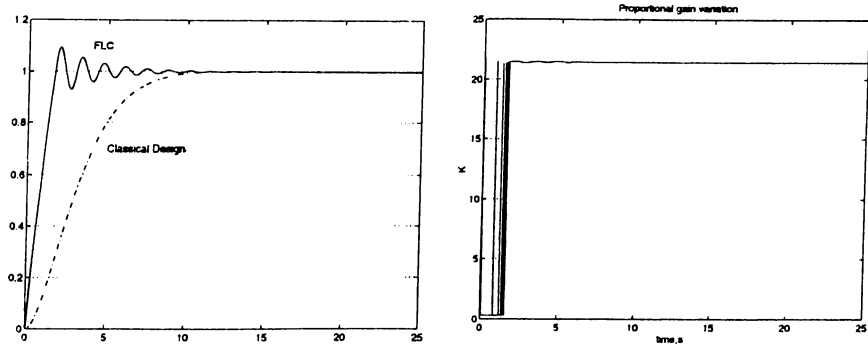


Figure 5.8: (a) The step responses with two different controller for the system with disturbed parameter (b) The waveform for the proportional gain.

This design is tested on a system whose transfer function is given as

$$P(s) = \frac{Y(s)}{U(s)} = \frac{1}{s(s+1)}, \quad (5.11)$$

which can be assumed as the disturbed version of the system the design is made on. The simulation results are given in Fig.5.8. In this figure the dashed line corresponds to the system with constant gain. Here it is observed that the fuzzy controller is more insensitive to the parameter variations than the controller with constant value.

Chapter 6

Conclusion

In this thesis, fuzzy logic controller (FLC) design for tuning some parametric controllers are investigated.

In the fourth chapter, two designs have been realized by using heuristic rule generation, one for a PID controller and one for a lead-lag type of controller. The FLCs in these designs set the parameters of the PID and lead-lag controller on-line. The rules and the corresponding membership functions are constructed by observing the effect of the changes of the parameters on the overall performance. The simulation results of these controllers show better control performance relative to their classical design. The disadvantage of the heuristic design is that there is no analytical tool to construct the rule base. The designer should collect as much information as possible on the system and utilize them to construct the rule base and the related data base including the membership functions, set operations and inference engine.

In the design given in the fifth chapter, Fuzzy c-Means clustering algorithm is used to extract the rules and the membership functions from the input-output data of the system. The difficulties of this method is the need for an optimal input to the system. Even for a linear system the problem of tracking transforms the system to a nonlinear one. Solution methods may be numerically integrating the differential equations or applying the dynamic

programming method. Both of them have some disadvantages. In numerical integration of the system equations there might be singular points which cause the integral to blow up. In dynamic programming due to the discretization of the state space and input space the solution may not be the optimal solution. The discretization of the input domain gives a set of points in the input space. This set must be chosen so that the desired states can be reached in a feasible number of stages. Apart from these disadvantages the dynamic programming gives a reasonable suboptimal solution.

Simulation results showed that better controller performance can be achieved by FLCs in comparison with the classical design methods. The heuristic rule generation gives better tracking performance compared to the rule generation using the Fuzzy c-Means algorithm. One of the utilization of fuzzy c-means algorithm may be in system identification using the input and output data of the system.

REFERENCES

- [1] B. Kosko,. *Neural Networks and Fuzzy Systems*. Englewood Cliffs, NJ, Prentice Hall, 1992.
- [2] Chuen Chien Lee, “Fuzzy logic in control systems: Fuzzy logic controller-part1,part2,” *IEEE Trans. on System, Man, and Cybernetics*, vol. 20, pp. 404–432, March/April 1990.
- [3] Hao Ying, “Practical design of nonlinear fuzzy controllers with stability analysis for regulating processes with unknown mathematical models,” *Automatica*, vol. 30, pp. 1185–1195, 1994.
- [4] Zhen-Yu Zhao, Masayoshi Tomizuka, Satoru Isaka, “Fuzzy gain scheduling of PID controller,” *IEEE Transactions On Systems, Man, and Cybernetics*, vol. 23, pp. 1392–1398, Sep/Oct 1993.
- [5] Jihong Lee, “On methods for improving performance of pi-type fuzzy logic controllers,” *IEEE Transactions On Fuzzy Systems*, vol. 1, pp. 298–301, Nov 1993.
- [6] Zadeh L.A., “Fuzzy sets,” *Information and Control*, vol. 8, pp. 338–353, 1965.
- [7] Dimiter Driankov, Hans Hellendoom, Michael Reinfrank,. *An Introduction to Fuzzy Control*. Springer-Verlag, 1993.
- [8] Witold Pedrycz,. *Fuzzy Control and Fuzzy Systems*. Research Studies Press Ltd., 2 edition, 1993.
- [9] Li-Xin Wang. *Adaptive Fuzzy Systems and Control, Design and Stability Analysis*. Prentice Hall, 1994.

- [10] Mohammad R. Emami, I. Burhan Turksen, Andrew A. Goldenberg,. “An improved fuzzy modeling algorithm, part 1: Inference mechanism,”. Appeared in Proceedings of NAFIPS’96, June 20-21 1996 UC. Berkley, USA.
- [11] Mohammad R. Emami, I. Burhan Turksen, Andrew A. Goldenberg,. “An improved fuzzy modeling algorithm, part 2: System identification,”. Appeared in Proceedings of NAFIPS’96, June 20-21 1996 UC. Berkley, USA.
- [12] J. G. Zeigler and N. B. Nichols, “Optimum settings for automatic controller,” *Trans. ASME*, vol. 64, pp. 759–765, 1942.
- [13] Benjamin C. Kuo,. *Automatic Control System*. Prentice-Hall, 1987.
- [14] Donald E. Kirk,. *Optimal Control Theory, An Introduction*. Prentice-Hall Inc., 1970.
- [15] Nikhil R. Pal and James C. Bezdek, “On cluster validity for th fuzzy c-means model,” *Trans. on Fuzzy Systems*, vol. 3, pp. 370–379, Aug 1995.
- [16] Keller J. M., Gray M. R., Givens J. A., “A fuzzy k-nearest algorithm,” *IEEE Trans. Systems, Man, and Cybernetics*, vol. SMC-15, pp. 580–585, 1985.